

Dytran 2022.2

Reference Manual

HEXAGON

Americas
5161 California Ave. Suite 200
University Research Park, Irvine, CA 92617
Telephone:(714) 540-8900
Toll Free Number: 1 855 672 7638
Email: americas.contact@mscsoftware.com

Europe, Middle East, Africa
Am Moosfeld 13
81829 Munich, Germany
Telephone: (49) 89 431 98 70
Email: europe@mscsoftware.com

Japan
Shinjuku First West 8F
23-7 Nishi Shinjuku
1-Chome, Shinjuku-Ku
Tokyo 160-0023, JAPAN
Telephone: (81) (3)-6911-1200
Email: MSCJ.Market@mscsoftware.com

Asia-Pacific
100 Beach Road
#16-05 Shaw Tower
Singapore 189702
Telephone: 65-6272-0082
Email: APAC.Contact@mscsoftware.com

Worldwide Web
www.hexagon.com

Support
https://simcompanion.hexagon.com/

Disclaimer

This documentation, as well as the software described in it, is furnished under license and may be used only in accordance with the terms of such license.

Hexagon reserves the right to make changes in specifications and other information contained in this document without prior notice. The concepts, methods, and examples presented in this text are for illustrative and educational purposes only, and are not intended to be exhaustive or to apply to any particular engineering problem or design. Hexagon assumes no liability or responsibility to any person or company for direct or indirect damages resulting from the use of any information contained herein.

User Documentation: Copyright ©2022 Hexagon AB and/or its subsidiaries. All Rights Reserved.

This notice shall be marked on any reproduction of this documentation, in whole or in part. Any reproduction or distribution of this document, in whole or in part, without the prior written consent of Hexagon is prohibited. The concepts, methods, and examples presented in this document are for illustrative and educational purposes only and are not intended to be exhaustive or to apply to any particular engineering problem or design. THIS DOCUMENT IS PROVIDED ON AN “AS-IS” BASIS AND ALL EXPRESS AND IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.
This software may contain certain third-party software that is protected by copyright and licensed from Hexagon suppliers. Additional terms and conditions and/or notices may apply for certain third party software. Such additional third party software terms and conditions and/or notices may be set forth in documentation and/or at http://www.mscsoftware.com/thirdpartysoftware (or successor website designated by Hexagon from time to time).

The Hexagon logo, Hexagon, MSC Software logo, MSC, MSC Nastran, Marc, Patran, Dytran, and Laminate Modeler are trademarks or registered trademarks of Hexagon AB and/or its subsidiaries in the United States and/or other countries.

NASTRAN is a registered trademark of NASA. All other trademarks are the property of their respective owners.
Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software) and DFARS 227.7202 (Commercial Computer Software and Commercial Computer Software Documentation), as applicable.

DT:V2022.2:Z:Z:Z:DC-REF-PDF

Documentation Feedback

At Hexagon, we strive to produce the highest quality documentation and welcome your feedback. If you have comments or suggestions about our documentation, please write to us.

Please include the following information with your feedback:

- Document name
- Release/Version number
- Chapter/Section name
- Topic title (for Online Help)
- Brief description of the content (for example, incomplete/incorrect information, grammatical errors, information that requires clarification or more details and so on.)
- Your suggestions for correcting/improving documentation

You may also provide your feedback about Hexagon documentation by taking a short 5-minute survey.

Note:	The above mentioned e-mail address is only for providing documentation specific feedback. If you have any technical problems, issues, or queries, please contact Technical Support.

Contents

Dytran Reference Manual

1 Introduction

Overview **2**

Similarity with MSC Nastran **3**

Input. 3

Loading 5

2 File Management Statements (FMS)

Overview **8**

Summary **8**

Prestress Analysis 8

New Analyses. 8

Restart Control 8

User Code 8

File Selection 9

FMS Descriptions **9**

BULKOUT — Prestress Grid Point Output. 10

CONNECT — User Defined Service and Group Evaluator Data 11

EULINIT — Imports an Euler archive from a previous run. 12

IMMFILE — Initial Metric Method File Section 14

NASTDISP — Prestress MSC Nastran Displacement File 15

NASTINP — Prestress MSC Nastran Solution File 16

NASTOUT — MSC Nastran Input File for Prestress Analysis. 17

PRESTRESS — Prestress Analysis. 18

RESTART — Restarts a Previous Run 19

RSTBEGIN — Restart Time Step 20

RSTFILE — Restart File Section. 21

SAVE — Interval Between Saving an Output File 22

SOLINIT — Specify an Initial Solution File from Prestress Analysis. 23

SOLUOUT — Specifies the Output Solution File from a Prestress Analysis 24

START — Primary Analysis 25

TYPE — Type of Output File 26

3 Executive Control Statements

Overview **28**
Executive Control Summary **28**
Executive Control Descriptions **28**
CEND — Terminates the Executive Control Section 29
MEMORY-SIZE — Definition of Memory Usage 30
TIME — Selects the Maximum CPU Time 31

4 Case Control Commands

Overview **36**
Case Control Summary **36**
Analysis Control 36
Data Selection 36
Output Control 36
Output Selection – Entity Specification 36
Output Selection – Variable Specification 37
Output Frequency 37
User-Defined Output 37
Input File Control 38
Miscellaneous 38
Case Control Descriptions **38**
ACC — Accelerometer Output 39
CHECK — Data Check 40
CMARKOUT — Marker Data to be Output 41
CMARKS — CMARKS to be Output 42
COG — Center of Gravity Calculation 43
CONTOUT — Contact Surface Data to be Output 44
CONTS — Contact Surfaces to be Stored 45
CORDDEF — Coordinate System for Deformation Output 46
CPLSBOUT — Coupling Subsurface Data to be Output 47
CPLSOUT — Coupling Surface Data to be Output 48
CPLSUBS — Coupling Subsurfaces to be Output 49
CPLSURFS — Coupling Surfaces to be Output 50
CSECS — Cross Sections to be Output 51
CSOUT — Cross-section Data to be Output 52
EBDOUT — Eulerian Boundary Data to be Output 53
EBDS — Eulerian Boundaries to be Output 54
ELEMENTS — Elements to be Output 56
ELEXOUT — User-defined Element Output 57
ELOUT — Element Data to be Output 58
ENDSTEP — Final Time Step 59

ENDTIME — Analysis Termination Time 60
GBAGOUT — Gas Bag Data to be Output 61
GBAGS — Gas Bags to be Output 62
GPEXOUT — User-defined Grid Point Output 63
GPOUT — Grid Point Data to be Output 64
GRIDS — Grid Points to be Output 65
HIC — Head Injury Criteria Calculation 66
INCLUDE — Starts Reading of a New File 68
MATBOUT — MATBX Data to be Output 69
MATBX — Boxes to be Output 70
MATOUT — Material Data to be Output 71
MATS — Materials to be Output 72
PARAM — Parameter Specification 73
RBOUT — Rigid Body Data to be Output 74
RELOUT — Rigid Ellipsoid Data to be Output 75
RELS — Rigid Ellipsoids to be Output 76
RIGIDS — Rigid Bodies to be Output 77
SET — Set Definition 78
SETC — List of Names 80
SGAUGES — Surface Gauges to be Stored 81
SPC — Single Point Constraint Set Selection 82
STEPS — Time Steps at which Data is Written 83
SUBSOUT — Subsurface Data to be Output 84
SUBSURFS — Subsurfaces to be Stored 85
SURFACES — Surfaces to be Stored 86
SURFOUT — Surface Data to be Output 87
TIC — Transient Initial Condition Selection 88
TIMES — Times at which Data is Written 89
TITLE — Output Title 90
TLOAD — Transient Load Selection 91
WALLOUT — Wall Data to be Output 92
WALLS — Walls to be Output 93

5 Bulk Data Entry Descriptions

Overview 106
Format of Bulk Data Entries 106
Free Field Format 106
Bulk Data Summary 108
Geometry 108
Lagrangian Elements 110
Eulerian Elements 111
Constitutive Models 111

Rigid Bodies . . . 113
ATB Interface . . . 113
Lagrangian Constraints . . . 114
Lagrangian Loading . . . 115
Eulerian Loading and Constraints . . . 116
Euler/Lagrange Coupling . . . 118
Miscellaneous . . . 119
Bulk Data Descriptions . . . 120
$ — Comment . . . 121
ACTIVE — Activate Elements and Interaction . . . 122
ALE — Arbitrary Lagrange Eulerian (ALE) Interface . . . 124
ALEGRID — Eulerian Grid Point Motion Definition . . . 125
ALEGRID1 — Eulerian Grid Point Motion Definition . . . 128
ATBACC — Acceleration Field Applied to ATB Segments . . . 131
ATBJNT — Interface to ATB Joints . . . 132
ATBSEG — Interface to ATB Segments . . . 135
BARLT89 — Sheet-Metal Material . . . 137
BARLT00 — Barlat YLD-2000 material . . . 142
BEGIN BULK — The Beginning of the Bulk Data . . . 146
BIAS — Bias Definition . . . 147
BJOIN — Breakable Join . . . 150
BODYFOR — Body Force Loading . . . 154
BODYFR1 — Body Force for Eulerian Regions . . . 156
BOX — Defines the Shape of a Box . . . 158
BOX1 — Defines the Shape of a General BOX . . . 159
CBAR — Bar Element Connection . . . 160
CBEAM — Beam-Element Connectivity . . . 162
CBUSH — Generalized Spring-and-Damper Connection . . . 164
CDAMP1 — Damper Connectivity . . . 166
CDAMP2 — Linear Damper . . . 168
CELAS1 — Scalar-Spring Connection . . . 170
CELAS2 — Scalar-Spring Property and Connection . . . 172
CFACE — Face of an Element . . . 174
CFACE1 — Face of an Element . . . 176
CHEXA — Element with Eight Grid Points . . . 177
CIFHEX — Solid Interface Cohesive Zone Modeling Element . . . 179
CMARKB2 — Two-noded Marker Connectivity Definition . . . 180
CMARKN1 — One-noded Marker Connectivity Definition . . . 181
COHFRIC — Cohesive Friction . . . 182
CONM2 — Concentrated Grid Point Mass and/or Inertia . . . 183
CONTACT — Contact Surface . . . 184
CONTFORC — Contact Force Definition Using Force-Deflection Curves . . . 204
CONTINI — Contact Initialization for In-Plane Folded Air Bags . . . 206
CONTREL — Contact with Rigid Ellipsoids . . . 208
CORD1C — Cylindrical Coordinate System Definition, Form 1 . . . 209

CORD1R — Rectangular Coordinate System Definition, Form 1 211
CORD1S — Spherical Coordinate System Definition, Form 1 213
CORD2C — Cylindrical Coordinate System Definition, Form 2 215
CORD2R — Rectangular Coordinate System Definition, Form 2 217
CORD2S — Spherical Coordinate System Definition, Form 2 219
CORD3R — Moving Rectangular Coordinate System Definition, Form 1 221
CORD4R — Moving Rectangular Coordinate System Definition, Form 2 223
CORDROT — Corotational-frame Definition 225
COUCOHF — Cohesive Friction Model to be Used for a COUPLE Entry 226
COUHTR — Heat Transfer Model to be Used with COUPLE Entry 227
COUINFL — Inflator Model to be Used with COUPLE Entry 229
COUOPT — Coupling Options 231
COUP1FL — Coupling Surface Failure 233
COUP1INT — Coupling Surface Interaction 234
COUPLE — General Euler-Lagrange Coupling Surface 235
COUPLE1 — Euler-Lagrange Coupling Surface 239
COUPOR — Coupling Porosity 241
CPENTA — Solid Element with Six Grid Points 246
CQUAD4 — Quadrilateral Element Connection 247
CROD — Rod Element Connection 249
CSEG — Segment of a Contact Surface or Coupling Surface 250
CSPR — Spring Connection 251
CTETRA — Solid Element with Four Grid Points 252
CTRIA3 — Triangular Element Connection 253
CVISC — Damper Connection 255
CYLINDER — Defines the Shape of a Cylinder 256
DAREA — Concentrated Load or Enforced Motion 257
DETSPH — Spherical Detonation Wave 258
DMAT — General Constitutive Model 259
DMATEL — Isotropic Elastic Material Properties 261
DMATEP — Elastoplastic Material Properties 263
DMATOR — Orthotropic Elastic Material Properties 264
DRAWBD — Drawbead definition 267
DRAWBD1 — Drawbead definition 269
DYMAT14 — Soil and Crushable Foam Material Properties 271
DYMAT24 — Piecewise Linear Plasticity Material 274
DYMAT25 — Cap Material Model 276
DYMAT26 — Orthotropic Crushable Material Model 278
ENDDATA — Terminates the Input Data 282
EOSDEF — Deflagration 283
EOSEX — User-defined Equation of State 286
EOSGAM — Gamma Law Gas Equation of State 288
EOSIG — Ignition and Growth Equation of State 290
EOSJWL — JWL Explosive Equation of State 296

EOSMG — Mie-Gruneisen Equation of State 298
EOSNA — Noble-Abel equation of state 300
EOSPOL — Polynomial Equation of State 302
EOSTAIT — Tait Equation of State 305
FABRIC — Woven Fabric Material 307
FAILEST — Maximum Equivalent Stress and Minimum Time Step Failure Model 311
FAILEX — User Failure Subroutine 312
FAILEX1 — Extended User Failure Subroutine 313
FAILEX2 — User Failure Subroutine 314
FAILJC — Johnson-Cook Failure Model 315
FAILMES — Maximum Equivalent Stress Failure Model 317
FAILMPS — Maximum Plastic Strain Failure Model 318
FAILPRS — Maximum Pressure Failure Model 319
FAILSDT — Maximum Plastic Strain and Minimum Time-Step Failure Model 320
FFCONTR — Closed Volume Intended for Fluid Filled Containers 321
FLOW — Flow Boundary Condition 323
FLOWC — Cyclic Flow Boundary Condition 325
FLOWCDR — Cyclic Flow Boundary Condition for All Faces in Certain Direction 328
FLOWCSQ — Cyclic Flow Boundary Condition using a Square Definition 331
FLOWDEF — Default Flow Boundary 334
FLOWDIR — Flow Boundary Condition for all Eulerian Boundary Faces in a Specific Direction 336
FLOWEX — User-defined Flow Boundary 338
FLOWSQ — Flow Boundary Condition using a Square Definition 340
FLOWT — Time-dependent Flow Boundary 343
FLOWTSQ — Time dependent Flow Boundary 346
FLOWXDR — Eulerian Boundary Faces in Specific Direction 350
FLOWXSQ — Eulerian Boundary Faces Inside Specified Square 352
FOAM1 — Foam Material Properties 354
FOAM2 — Foam Material Properties 356
FORCE — Concentrated Load or Velocity 359
FORCE1 — Follower Force, Form 1 361
FORCE2 — Follower Force, Form 2 362
FORCE3 — Grid Point Velocity Definition 363
FORCEEX — User-defined Enforced Motion at Grid Points 365
GBAG — Gas Bag Pressure Definition 366
GBAGC — Gas Bag Connection 373
GBAGCOU — General Coupling to Gas Bag Switch 376
GBAGHTR — Heat Transfer Model to be used With GBAG Entry 377
GBAGINFL — Inflator Model to be used with GBAG Entry 379
GBAGPOR — Gas Bag Porosity 381
GRAV — Gravity Field 384
GRDSET — Grid Point Default 385
GRID — Grid Point 386
GROFFS — Grid Point Offset 387
HGSUPPR — Hourglass Suppression Method 388

HTRCONV — Air Bag Convection . . . 391
HTRRAD — Air Bag Radiation . . . 392
HYDSTAT — Hydrostatic Preset of Density in Euler Elements . . . 393
IGNORE — Ignore a Set of Euler Elements . . . 395
INCLUDE — Starts Reading a New Input File . . . 396
INFLATR — Air Bag Inflator Model . . . 397
INFLATR1 — Air Bag Inflator Model . . . 399
INFLCG — Airbag Cold Gas Inflator Model . . . 401
INFLFRAC — Hybrid Inflator Gas Fraction Definition . . . 403
INFLGAS — Inflator Gas Definition . . . 404
INFLHYB — Hybrid Inflator Model . . . 406
INFLHYB1 — Hybrid Inflator Model . . . 407
INFLTANK — Air Bag Tanktest Inflator Model . . . 408
INITGAS — Gas Bag or Coupling Surface Initial Gas Fraction Definition . . . 410
JOIN — Hinge-type Join of Six DOF Grid Points with Three DOF Grid Points . . . 411
KJOIN — Kinematic Join of Six DOF Grid Points with Three DOF Grid Points . . . 412
MAT1 — Material Property Definition, Form 1 . . . 413
MAT2 — Anisotropic Material for Shells . . . 414
MAT8 — Orthotropic Elastic Material Properties . . . 415
MAT8A — Orthotropic Failure Material Properties . . . 416
MATINI — Eulerian Initialization Surface . . . 421
MATRIG — Rigid-Body Properties . . . 423
MCOHE — Interface Cohesive Zone Modeling Element Material Properties . . . 426
MESH — Mesh Generator . . . 430
MOMENT — Concentrated Moment or Enforced Motion . . . 439
MOMENT1 — Follower Moment, Form 1 . . . 440
MOMENT2 — Follower Moment, Form 2 . . . 441
NASINIT — MSC Nastran Initialization . . . 442
PARAM — Parameter . . . 443
PBAR — Simple Beam Property . . . 444
PBCOMP — Beam Alternate Form of PBEAM . . . 445
PBEAM — Beam Property . . . 447
PBEAM1 — Beam Properties (Belytschko-Schwer) . . . 449
PBEAM1 — Beam Properties (Hughes-Liu) . . . 453
PBEAM1 — Beam Properties (Predefined Hughes-Liu Cross Sections) . . . 455
PBEAML — Beam Cross-Section Properties . . . 460
PBELT — Belt Property . . . 464
PBUSH — Generalized Spring-and-Damper Property . . . 466
PBUSHT — Nonlinear Force Deflection Spring Property . . . 467
PCOHE — Interface Cohesive Zone Modeling Element Properties . . . 472
PCOMP — Layered Composite Element Property . . . 473
PCOMPA — Additional Data for Layered Composite Element Property . . . 475
PDAMP — Linear Damper Properties . . . 477
PELAS — Elastic Spring Property . . . 478

PELAS1 — Nonlinear Elastic Spring with Hysteresis Property 479
PELASEX — UDS Properties of an Elastic Spring 480
PERMCPL — Permeability between Coupling Surfaces 482
PERMEAB — Air Bag Permeability 484
PERMGBG — Air Bag Permeability 486
PEULER — Eulerian Element Properties 487
PEULER1 — Eulerian Element Properties 489
PLOAD — Pressure Loads on the Face of Structural Elements 490
PLOAD4 — Pressure Loads on the Face of Structural Elements 491
PMARKER — Property Definition of a Marker Element 494
PMINC — Constant Spallation Model 495
POREX — User-defined Porosity Model Specified by a User Subroutine 497
PORFCPL — Flow Between Two Coupling Surfaces Through a Hole 498
PORFGBG — Flow Between Two Air Bags Through a Hole 499
PORFLCPL — Flow Between Two Coupling Surfaces Through a Large Hole 500
PORFLGBG — Flow Between Two Air Bags Through a Large Hole 501
PORFLOW — Porous Flow Boundary 503
PORFLOWT — Time-dependent Porous Flow Boundary 505
PORHOLE — Holes in Air Bag Surface 508
PORHYDST — Porous Flow Boundary 509
PORLHOLE — Large Hole in Air Bag Surface 510
PROD — Rod Property 512
PSHELL — Shell-Element Properties 513
PSHELL1 — Shell-Element Properties 515
PSOLID — Lagrangian Solid-Element Properties 518
PSPR — Linear-Elastic Spring Properties 520
PSPR1 — Nonlinear-Elastic Spring with Hysteresis Properties 521
PSPREX — UDS Properties of a Linear Elastic Spring 522
PVISC — Linear-Damper Properties 523
PVISC1 — Nonlinear Damper Properties 524
PVISCEX — UDS Properties of a Linear Viscous Damper 525
PWELD — Spotweld Property 526
PWELD1 — Skin-Stringer Delamination 529
PWELD2 — Sandwich Structure Delamination 533
RBC3 — Rigid-Body Constraint 536
RBE2 — Rigid-Body Element 538
RBHINGE — Rigid Body Hinge 540
RCONN — Rigid Connection 541
RCONREL — Rigid Connection with Rigid Ellipsoids 544
RELEX — External Definition of a Rigid Ellipsoid 545
RELLIPS — Rigid Ellipsoid 546
RFORCE — Rotational Force Field 547
RIGID — Rigid Surface 548
RJCYL — Cylindrical-Joint Constraint Between Rigid Bodies 550
RJPLA — Planar-Joint Constraint Between Rigid Bodies 552

RJREV — Revolute-Joint Constraint Between Rigid Bodies 554
RJSPH — Spherical-Joint Constraint Between Rigid Bodies. 556
RJTRA — Translational-Joint Constraint Between Rigid Bodies 558
RJUNI — Universal-Joint Constraint Between Rigid Bodies 560
RUBBER1 — Mooney-Rivlin Rubber Material 562
RUBBER2 — Ogden Rubber Material 564
SECTION — Cross Section. 566
SET1 — Set of Numbers 567
SETC — List of Names 568
SETTING — Application-Sensitive Defaults 569
SHEETMAT — Sheet-Metal Material 571
SHREL — Elastic Shear Model. 576
SHREX — User-specified Shear Model 577
SHRLVE — Isotropic Linear Viscoelastic Shear Model 578
SHRPOL — Polynomial Shear Model 580
SPC — Single-Point Constraint 581
SPC1 — Single-Point Constraint 582
SPC2 — Single-Point Constraint 583
SPC3 — Single-Point Constraint 585
SPHERE — Defines the Shape of a Sphere 587
SUBSURF — Multifaceted Subsurface 588
SURFACE — Multifaceted Surface 590
TABFILE — Text File Defined Function. 592
TABLED1 — Table. 593
TABLEEX — User-defined Function 595
TIC — Transient Initial Velocities of Grid Points 596
TIC1 — Transient Initial Velocities of Grid Points 597
TIC2 — Transient Initial Velocities of Grid Points 598
TIC3 — General Form of Transient Initial Velocities of Grid Points 599
TICEEX — User-defined Transient Initial Conditions of Elements 600
TICEL — Transient Initial Conditions of Elements. 601
TICEUL — Transient Initial Conditions of Eulerian Regions 602
TICGEX — User-defined Transient Initial Conditions of Grid Points 604
TICGP — Transient Initial Conditions for Grid Points. 605
TICVAL — Transient Initial Condition Set 606
TLOAD1 — Transient Dynamic Load 608
TLOAD2 — Transient Dynamic Load, Form 2. 610
VISCDMP — Dynamic Relaxation. 612
WALL — Lagrangian Rigid Wall 615
WALLDIR — Wall Boundary Condition for all Eulerian Boundary Faces in a Specific Direction 617
WALLET — Barrier for Eulerian Transport 618
YLDEX — User-defined Yield Model. 619
YLDHY — Hydrodynamic Yield Model 620
YLDJC — Johnson-Cook Yield Model 621

YLDMC — Mohr-Coulomb Yield Model . . . 623
YLDMSS — Multisurface Yield Model for Snow . . . 624
YLDPOL — Polynomial Yield Model . . . 626
YLDRPL — Rate Power Law Yield Model . . . 627
YLDSG — Steinberg-Guinan Yield Model . . . 628
YLDTM — Tanimura-Mimura Yield Model . . . 630
YLDVM — von Mises Yield Model . . . 632
YLDZA — Zerilli-Armstrong Yield Model . . . 635

6 Parameters

Overview . . . 643
Parameters Summary . . . 643
Contact Control . . . 643
Coupling Subcycling . . . 643
Blending Control . . . 643
Time-Step Control . . . 643
Limits . . . 644
Restart Control . . . 644
ALE Motion Control . . . 644
Hourglass Suppression Control . . . 644
Miscellaneous . . . 644
Material Parameter Control . . . 645
Shell Options . . . 645
Dynamic Relaxation . . . 645
Airbag Control . . . 645
ATB Positioning . . . 646
Output Control . . . 646
Prestressing Analysis . . . 646
Parallelization . . . 647
Parameter Descriptions . . . 648
ALEITR — Number of ALE Mesh Iterations . . . 649
ALETOL — Tolerance at ALE Interface . . . 650
ALEVER — ALE Volume Computation Method . . . 651
ATB-H-OUTPUT — Write ATB Output to Dytran Time-History Files . . . 652
ATBAOUT — Output Frequency to Main Output File of ATB . . . 653
ATBSEGCREATE — Create Grids and Elements for ATBSEG . . . 654
ATBTOUT — Output Frequency to Time-History Files of ATB . . . 655
AUTHINFO — Licensing Information Control . . . 656
AUTHQUEUE — Licensing Queuing Control . . . 657
AUTOCOUP — Automatic Coupling . . . 658
AXIALSYM — Axial Symmetric Analyses . . . 661
AXREMAP — 2-D Axial Symmetric Euler Archive Remap . . . 663
BULKL — Linear Bulk Viscosity Coefficient . . . 664

BULKQ — Quadratic Bulk Viscosity Coefficient . . . 665
BULKTYP — Bulk Viscosity Type . . . 666
CFULLRIG — Converts 123456 Constraints to FULLRIG on RBE2 Entries . . . 667
CLUFLIM — Limiter of Volume Stain Rate for Clumps . . . 668
CLUMPENER — Switch for Kinetic Energy Calculation Scheme of Blended Clumps . . . 669
COHESION — Cohesion for Coulomb Friction . . . 670
COMPSYS — PCOMP Shell Element System Definition . . . 671
COMPTCK — PCOMP Shell-Thickness Modification Option . . . 672
CONM2OUT — CONM2 Summary Output . . . 673
CONTACT — Sets Defaults for CONTACT . . . 674
COSUBCYC — Growth of Subcycling Interval in Coupling . . . 678
COSUBMAX — Subcycle Limit in Euler/Lagrange Coupling . . . 679
COSUBMAXT — Time-dependent Subcycle Limit in Euler/Lagrange Coupling . . . 680
COUFRIC — Coupling Surface Friction for Nonmetallic Eulerian Solids . . . 681
DELCLUMP — Delete Clump Fraction . . . 682
DMPOPT — Sets DMP option . . . 683
DUMSEGS — Creating dummy segments for the auto coupling approach . . . 684
ELDLTH — Show List of Lagrangian Elements with Time Step in Ascending Order . . . 685
ERRUSR — Redefinition of Severity and Number of Error Message Prints . . . 686
EULER-BOUNDARY — Euler Boundary Treatment . . . 687
EULERCUB — Cube Sorting for Euler Domains . . . 688
EULERPR — Euler Cube Partition Control . . . 689
EULSTRESS — Stress Update Method . . . 691
EULTRAN — Switch for the Euler Transport Scheme of the Multi-material Solver and the Single Material Strength Solver . . . 692
EUSUBCYC — Growth or Subcycling Interval in Euler Computations . . . 693
EUSUBMAX — Subcycle Limit in Euler Solver . . . 694
FAILDT — Element Time-step Based Failure Model . . . 696
FAILOUT — Failed Element Output Parameter . . . 697
FALENEG — Eroding energy control parameter . . . 698
FASTCOUP — Fast Coupling Algorithm . . . 699
FBLEND — Blend Fraction . . . 702
FLOW-METHOD — Flow-Method Between Two Euler Domains Across Open Areas in Coupling Surfaces . 703
FMULTI — Multimaterial Overflow Array Parameter . . . 706
GEOCHECK — Define Geometry Consistency Check . . . 707
GRADED-MESH — Glue Sets of Euler Elements . . . 708
HGCMEM — Shell Membrane Hourglass Damping Coefficient Parameters . . . 709
HGCOEFF — Hourglass Damping Coefficient . . . 710
HGCSOL — Solid Hourglass Damping Coefficient . . . 711
HGCTWS — Shell Twisting Hourglass Damping Coefficient . . . 712
HGCWRP — Shell Warping Hourglass Damping Coefficient . . . 713
HGSHELL — Shell Hourglass Suppression Method . . . 714
HGSOLID — Solid Hourglass Suppression Method . . . 715
HGTYPE — Hourglass Suppression Method . . . 716

HICGRAV — Gravity Used by HIC Calculations . . . 717
HVLFAIL — Failure at Hydrodynamic Volume Limit . . . 718
HYDROBOD — Hydrodynamic Body Force . . . 719
IEEE — IEEE Binary Data Output Format . . . 720
IGNFRCER — Ignores Warnings . . . 721
IMM — Initial Metric Method Formulation . . . 722
INFO-BJOIN — List the Generated BJOINs and Spotwelds . . . 724
INISTEP — Initial Time Step . . . 725
INITFILE — Method of Initialization Definition . . . 726
INITNAS — Defines the Type of Displacement Initialization File . . . 728
JWLDET — Blast Wave Ignition of One to Another . . . 730
LAGPR — Lagrangian Partitioing Control . . . 731
LIMCUB — Contact Cube Sort Algorithm . . . 732
LIMITER — Euler Solver Scheme . . . 733
MATRMERG — Merges MATRIG and RBE2-FULLRIG Assemblies . . . 734
MATRMRG1 — Merges MATRIG and RBE2-FULLRIG Assemblies . . . 735
MAXSTEP — Maximum Time Step . . . 736
MESHELL — Mesh Density for Covering Ellipsoids . . . 737
MESHPLN — Mesh Density for Covering Rigid Planes . . . 738
MICRO — Micro-zoning Parameter . . . 739
MINSTEP — Minimum Time Step . . . 740
MIXGAS — Controls Updating of Gas Fractions . . . 741
NASIGN — Echo Ignored Data Entries . . . 742
NZEROVEL — Auto Constrain Failed Nodes . . . 743
OLDLAGTET — Uses Collapsed Hexahedron Scheme for CTETRA . . . 744
OLDSHPL — Uses old plastic calculation for shell elements . . . 745
PARALLEL — Parallel Execution Information . . . 746
PLCOVCUT — Pressure Cut Off Time . . . 747
PMINFAIL — Switches Failure at Spall Limit . . . 748
RBE2INFO — Lists MATRIG and RBE2 Grid Points . . . 749
RHOCUT — Global Density Cutoff Value . . . 750
RJSTIFF — Rigid-joint Stiffness . . . 751
RKSCHEME — Runge-Kutta Time-Integration Scheme . . . 752
ROHYDRO — Density Cut-Off Value . . . 753
ROMULTI — Density Cut-Off Value . . . 754
ROSTR — Density Cut-Off Value . . . 755
RSTDROP — Type of Elements Dropped at Restart . . . 756
SCALEMAS — Mass Scaling Definition . . . 757
SHELLFORM — Sets the Default of the Shell Formulation . . . 759
SHELLKEY — Sets the KEYHOFF Shell Formulation . . . 760
SHELMSYS — Shell Element System Definition . . . 761
SHPLAST — Type of Plane-Stress Plasticity for Shells . . . 762
SHSTRDEF — Composite Shell Stress and Strain Output Definition . . . 763
SHTHICK — Shell-Thickness Modification Option . . . 764
SLELM — Store Shell Sublayer Variables . . . 765

SMF — Setting for sheet metal forming simulation . . . 766
SMP, ,BATCHSIZE — SMP Batchsize and CPULOOP Definition . . . 768
SMP, ,CPUINFO — SMP CPU Information . . . 770
SMSOPT — Time step scaling factor for selective mass scaling . . . 771
SNDLIM — Sound Speed Minimum Value . . . 773
SPREMAP — 1-D Spherical Symmetric Euler Archive Remap . . . 774
SPHERSYM — Spherical Symmetric Analyses . . . 775
STEPFCT — Time Step Scale Factor . . . 776
STEPFCTL — Time-step Scale Factor for Lagrangain Elements . . . 777
STRNOUT — Shell Sublayer Strain Output . . . 778
TOLCHK — Tolerance for Fast Coupling . . . 779
TOLFAC — Increase the Projection Tolerance for CONTACT at Initialization . . . 780
UGASC — Universal Gas Constant . . . 781
VARACTIV — Activation or Deactivation of Grid-Point, Element, or Face Variables . . . 782
VDAMP — Dynamic Relaxation Parameter . . . 784
VELCUT — Velocity Cutoff . . . 785
VELMAX — Maximum Velocity . . . 786
VELMAX1 — Maximum Translational and Angular Velocity . . . 787
VISCOPLAS — Use Overstress Formula to Update Strain-rate Dependent Plasticity . . . 789

7 User Defined Services

Overview . . . 791
Supported User Defined Subroutines . . . 791
Build Environment . . . 792
Path Descriptions . . . 792
Implementation with Templates . . . 792
Implementation without Templates . . . 793
Building the Service . . . 796
Environment Variables . . . 796
Using the Service . . . 796
Error Handling . . . 797
Utility Routines . . . 798
GET_TIME Routine — Mechanism to get Current Time . . . 798
GET_TIMESTEP Routine — Mechanism to get Current Timestep . . . 799
IGET_STEP Routine — Mechanism to get Current Step Number . . . 800
IGET_NSWRAP Routine — Mechanism to get Wrapping Condition . . . 801
IPUT_NSWRAP Routine — Mechanism to put Wrapping Condition . . . 802
Flow — User-defined Material Flow of an Euler Boundary or Coupling Surface in a Dytran Simulation . . . 803
InitOut — User-defined Initialization or Output in a Dytran Simulation . . . 811
Materials — User-defined Material Definition in a Dytran Simulation . . . 815

Properties — User-defined Property Definition in a Dytran Simulation 833
Loads — User-defined Loads in a Dytran Simulation. 840
Connections — User-defined Connections in a Dytran Simulation 843

8 Diagnostic Messages

A References

1 Introduction

- Overview 2
- Similarity with MSC Nastran 3

Overview

Input to Dytran™ takes the form of a data file where each line can contain up to 80 characters. The file contains all the information to define the analysis model and control the analysis.

The input to Dytran is similar, but not identical, to that for MSC Nastran® and Dyna. If you are familiar with MSC Nastran, learning to use Dytran will be very easy although you should note the areas in which the two programs differ. These differences are summarized in Similarity with MSC Nastran.

The input data is split into four main sections, which must be in the following order:

1. File Management Section (FMS)
2. Executive Control Section
3. Case Control Section
4. Bulk Data Section (Note that parameter options may appear at any location within the Bulk Data Section.)

The File Management Section contains information about the files used during the analysis and to control restarting. The Executive Control Section is not used often in Dytran, since the program does not have an Executive System like MSC Nastran.

The Case Control Section controls the analysis, specifies the type of input and output required, selects the constraints and loading from the Bulk Data, and allows you to control the way the analysis progresses. A discussion of the functions available in the Case Control Section and a detailed description of the commands that can be used is given in the Case Control section of this manual.

The Bulk Data Section contains all data necessary to define the model, the constraints, loading conditions, and initial conditions. Only one model can be defined in the input data, but several types of constraints and loading can be specified. The constraints and loading actually used in the analysis are selected in the Case Control Section. The Bulk Data Section is discussed in the this manual together with a detailed description of the entries.

The File Management, Executive Control, and Case Control Sections use a free-format input, which means that the data can appear anywhere on the line with individual items separated by commas or spaces. The Bulk Data Section can also be in free format and can optionally be in fixed format. In cases where additional precision is required, large format can be used, where each entry occupies two lines in the input file. Free, fixed, and large format can be mixed as needed in the input file on a line-by-line basis.

Comments can appear anywhere in the input file by placing a `$` at the start of the comment. A full description of the various input formats is given in Format of Bulk Data Entries.

The input data can be present in several separate files. In this case, you can use the `INCLUDE` command or entry, available in both the Case Control and Bulk Data Sections, to direct Dytran to read the appropriate file. The mechanism can be used to store the infrequently changed Bulk Data in one file, while the File Management, Executive Control, and Case Control Sections, which are usually modified more often, can be stored in another file.

Similarity with MSC Nastran

The input for Dytran is similar to the input for MSC Nastran, since the vast majority of the input for the two codes is identical. There are, however, a number of differences arising from the fundamental differences between the two programs, and the fact that there are features available in Dytran that are not available in MSC Nastran and vice versa.

Similarity to MSC Nastran has a number of advantages for anyone who works with both programs:

- You only need to learn one form of input.
- Models used for MSC Nastran analyses can be reused with minor modifications for Dytran.
- Dytran can be used with a wide range of modeling packages.

It is important to remember that MSC Nastran and Dytran are completely different programs even though they offer similar input. A CQUAD4 shell element in Dytran has nothing in common with the CQUAD4 shell element in MSC Nastran, since it differs in formulation, type of integration, and capabilities. Similarly, other features defined using the same entries do not necessarily behave in the same way. The solution method is different, so an identical analysis in MSC Nastran and Dytran can give slightly different results, although they will be within engineering accuracy.

Input

MSC Nastran has a wide range of facilities of which a number are not available in Dytran. Therefore, there are MSC Nastran entries that are not valid in Dytran.

The following entries are compatible with both codes:

Elements

CBAR	CHEXA
CBEAM	CQUAD4
CDAMP1	CROD
CDAMP2	CTETRA
CELAS1	CTRIA3
CELAS2	CVISC

Properties

PBAR	PROD
PBEAM	PSHELL
PCOMP	PSOLID
PDAMP	PVISC
PELAS	

Materials

MAT1	MAT8

Loads and Constraints

DAREA	MOMENT1
FORCE	MOMENT2
FORCE1	PLOAD
FORCE2	PLOAD4
GRAV	RFORCE
GRDSET	SPC
MOMENT	TIC

Coordinate Systems

CORD1C	CORD2C
CORD1R	CORD2R
CORD1S	CORD2S

Other Entries

CONM2	TITLE
GRID	TLOAD1
TABLED1	TLOAD5
TIME	

The FMS has the same purpose in both Dytran and MSC Nastran, but it is less important in Dytran since all the filenames are automatically defined. The FMS controls restarting and user-written subroutines as well as specification of the type of the output files.

The Executive Control Section exists but is rarely used since Dytran does not have an Executive System or DMAP.

The Case Control Section has the same function in both Dytran and MSC Nastran but uses different commands.

PARAM entries are used by Dytran but offer different options to those in MSC Nastran.

Dytran offers slightly greater flexibility in the way the input file can be defined, as listed below:

- Free format data can have more than eight characters.
- Continuation mnemonics do not have to be unique.
- Fixed and free format input can be freely mixed on a line-by-line basis.
- Real numbers can be entered as integers.

However, continuation lines must follow the entry that references them. If you intend on using both Dytran and MSC Nastran on a regular basis, use only those options that are available in both programs to avoid confusion and incompatibility.

Loading

Several of the entries used for static loading in MSC Nastran (such as FORCE, MOMENT, and PLOAD) are used for dynamic transient loading in Dytran. Instead of being referenced directly from Case Control, they are referenced from a TLOAD1 entry that gives the variation of the load with time.

The DAREA entry, used for transient loading in MSC Nastran, is also valid in Dytran.

2 File Management Statements (FMS)

- Overview 8
- Summary 8
- FMS Descriptions 9
- BULKOUT 10
- CONNECT 11
- EULINIT 12
- IMMFILE 14
- NASTDISP 15
- NASTINP 16
- NASTOUT 17
- PRESTRESS 18
- RESTART 19
- RSTBEGIN 20
- RSTFILE 21
- SAVE 22
- SOLINIT 23
- SOLUOUT 24
- START 25
- TYPE 26

Overview

The File Management Section (FMS) controls any file assignments that are required by Dytran. It also controls restarting. The FMS must be placed at the beginning of the input file, but the individual statements can be in any order within the FMS.

Most of the file assignments are made automatically by Dytran and cannot be changed by the user. The filenames used are machine dependent and are listed in the *Dytran Installation and Execution Guide*.

A summary of the statements available in the FMS is given below. Each statement is described in this chapter.

Summary

The following statements are valid in the FMS:

Prestress Analysis

PRESTRESS	Indicates a prestress analysis.
BULKOUT	Selects the file to which grid-point data is to be written.
NASTDISP	Selects the MSC Nastran displacement file to be used.
SOLUOUT	Selects a file to which solution data is to be written.

New Analyses

START	Indicates a new analysis.
NASTINP	Selects an MSC Nastran solution file from which Dytran is to be initialized.
SOLINIT	Selects a Dytran prestress solution file from which Dytran is to be initialized.
NASTOUT	Selects a file to which Dytran writes geometric and material data in MSC Nastran format.

Restart Control

RESTART	Indicates a restart of a previous analysis.
RSTFILE	Selects the restart file to be used.
RSTBEGIN	Selects the time step at which the calculation is to be restarted.

User Code

CONNECT	Defines the Simulation Component Architecture (SCA) User Defined Service (UDS) into Dytran via a new CONNECT SERVICE statement.

File Selection

TYPE	Defines the format of a file
SAVE	Defines the interval of saving an output file

FMS Descriptions

The format of the FMS statements is free field. In presenting the general formats for each statement, the following conventions are used:

- Uppercase letters should be typed as shown.
- Lowercase letters indicate that a value or option can be substituted.
- Brackets [ ] give a choice of different options.

The default value indicates the value that is used if no FMS command is present. The type column indicates the type of data you must supply. This can be I (Integer), R (Real), or C (Character). In addition, a range of permissible values may also be indicated. For example, I > 0 means that you must supply an integer that is greater than zero.

BULKOUT

Prestress Grid Point Output

Defines a file to which grid point data is written at the end of the prestress analysis.

Format and Example	Defaults
`BULKOUT = filename`	Required
`BULKOUT = GRID.DAT`	

Option	Meaning	Type
`filename`	The filename to be used	C

Remarks

1. The Bulk Data file filename contains only grid point data of the deformed geometry at the end of the prestress analysis. It can be used to construct an ALE mesh for the final transient dynamic analysis.
2. See NASTDISP, Prestress Analysis, and SOLUOUT in this chapter, and NASINIT, Chapter 5: Bulk Data Entry Descriptions in this manual.

CONNECT User Defined Service and Group Evaluator Data

Defines the newly introduced Simulation Component Architecture (SCA) User Defined Service (UDS) into Dytran via a new CONNECT SERVICE statement. Bulk Data entries such as EOSEX, MAT8A, etc. refer to this new service.

UDS Format:

```
CONNECT SERVICE <service_identifier> <service_name>
```

UDS Examples:

```
CONNECT SERVICE mysub SCA.MDSolver.Obj.Uds.Dytran.Materials
```

Creates a service identifier “mysub” which points to the “SCA.MDSolver.Obj.Uds.Dytran.Materials” service.

Option	Meaning	Type
group	Group name referenced by the GROUP field on the EOSEX, MAT8A, etc Bulk Data entries.	C
service_identifier	service_identifier is a name tag (8 characters long) which will be used to identify the service implementation to be used for a specific bulk data entry.	C
service_name	service_name is the name of service	C

Remarks

1. The process of enabling User Defined Services in Dytran consists of the following four steps:
   a. Creating the desired implementation for the User Defined Services in the form of dynamiclink libraries,
   b. Defining the location of the user defined service, service catalogue, service resource directory.
   c. Specifying the proper commands in the model to load the service,
   d. Identifying the elements that use the user supplied implementation.
2. In order to create a dynamic-link library suitable for usage with Dytran, a build environment is delivered to assist the user in building the library.

EULINIT

Imports an Euler archive from a previous run

Specifies an Euler archive used as input for a transient analysis. The Euler archive is mapped onto a set of Euler elements that can equal in size or either finer or coarser.

Format and Example	Default
`EULINIT, filename,CYCLE,MESH-ID`	Required

Option	Meaning	Type
`filename`	The filename to be used	C
`CYCLE`	Cycle number	
`MESH-ID`	The MESH ID of the target elements	See Remark 3.

Remarks

1. The target elements are the elements defined in the follow-up run and are the elements that will be initialized using the import archive.
2. Both defined and imported Euler elements need to be orthogonal in the global system
3. `MESH-ID` enables support for multiple Euler domains. If `MESH-ID` is not set the import archive will be mapped onto all Euler elements.
4. For multi-material Euler analyses with multiple Eulerian materials all material variables in the import archive require the material number. These material variables are `MASS`, `DENSITY`, `SIE`, `FMAT`. The required list of variables for a MMHYDRO run are: `MASSXX`, `SIEXX`, `FMATXX`, `XVEL`, `YVEL`, `ZVEL`. Here `XX` denotes the material number. FOR `MMSTREN` the variables `TXX`, `TYY`, `TZZ`, `TXY`, `TYZ`, `TZX`, `EFFPLS` have to be added. If the multi-material run uses only one Eulerian material then the material numbers can be left out.
5. For multi-material Euler analyses with EOSIG, the following has to be added for the `IG` materials: `MASS-EXX`, `MASS-PXX`, `RHO-EXX`, `RHO-PXX`, `IGBURNXX`, `FMAT-PXX`, `FMAT-PXX`, `SIE-EXX`, and `SIE-PXX`. Here `XX` denotes the material number.
6. For the single-material Hydro Euler solver the required list is `MASS`, `DENSITY`, `SIE`, `FMAT`, and `FVUNC`.
7. IF `FVUNC` is not included in the Import archive it is assumed that all elements in this archive are fully uncovered. It is allowed to import such an archive in a simulation with a coupling surface. In this follow up simulation the target elements can have uncover fractions different from one. In this case conservative quantities of imported elements are reduced by the uncover fraction of the target element. This is to avoid unwanted pressure increases. It simply means that any mass of the import archive that is located in the covered part of the target elements is thrown away. As a result not all mass in the import archive is mapped to the target elements. How much of the mass of the import archive is mapped is shown in the out file.

8. In the OUT file, a summary is shown of all variables that are mapped.
9. In the follow-up run, the cycle and time are taken from the import archive. The results of the first cycle of the follow-up run are determined from remapping only and has not gone trough an equation of state yet. This will happen in the next cycle.
10. If needed the remapping can be checked by doing only one additional cycle in the follow-up run with a quite small time step. Then, the follow-up OUT file shows two cycles and the results should be almost identical to the results of the import archive. Also, material summaries in the OUT file between first run and follow-up run should be identical. The only exceptions are the summaries of momentum, kinetic energy, and total energy per material. For these three quantities, only the total amounts will remain constant between first and follow-up run.
11. To remap a spherical symmetric or an axial symmetric Euler archive, the PARAMs SPREMAP and AXREMAP have to be used. Also, the remapping of a spherical symmetric Euler domain onto a 2-D axial symmetric Euler domain is supported.

IMMFILE

Initial Metric Method File Section

Defines the reference file to be used for the Initial Metric Method

Format and Example	Default
`IMMFILE = filename`	Required
`IMMFILE = flat.dat`	

Option	Meaning	Type
`filename`	The filename to be used for the Initial Metric Method. The file must exist in your runtime directory.	C

Remark

The Initial Metric Method is described in the *Dytran User's Guide* in Initial Metric Method for Air Bags.

NASTDISP

Prestress MSC Nastran Displacement File

Specifies an MSC Nastran displacement file to be used as input for the prestress analysis.

Format and Example	Defaults
`NASTDISP = filename`	Required
`NASTDISP = DISPLACE.DIS`	

Option	Meaning	Type
`filename`	The filename to be used	C

Remarks

1. The displacement file must be either in Patran format, formed by using `NASPAT` on the MSC Nastran `OUTPUT2` results file.
2. The default file type is MSC.XL format. This can be changed using `PARAM,INITNAS`.
3. See also the BULKOUT, Prestress Analysis, and SOLUOUT in this chapter, and NASINIT, Chapter 5: Bulk Data Entry Descriptions in this manual.

NASTINP

Prestress MSC Nastran Solution File

Specifies an MSC Nastran solution file from which Dytran is to be initialized via element stresses and grid-point displacements.

Format and Example	Defaults
`NASTINP = filename1, filename2`	Required
`NASTINP = ELEMENT. ELS, GRID. DIS`	

Option	Meaning	Type
`filename1, filename2`	The filenames to be used	C, C

Remarks

1. The stresses and displacement files are obtained by using `NASPAT` on the `OUTPUT2` results file from MSC Nastran.
2. Element stresses are defined in the material coordinate system.
3. It is recommended that the MSC Nastran geometrical problem setup be performed by Dytran for consistency (see NASTOUT).
4. This option causes Dytran to read a `MASS.DAT` file that is automatically generated by the NASTOUT File Management Section statement.

NASTOUT

MSC Nastran Input File for Prestress Analysis

Specifies that Dytran write out MSC Nastran input containing geometry and material definitions including material coordinate systems, if applicable.

Format and Example	Defaults
NASTOUT = filename	Required
NASTOUT = NASGEO.DAT	

Option	Meaning	Type
`filename`	The filename to be used.	C

Remark

The option causes a `MASS.DAT` file to be written containing the element initial masses. This file is read when the NASTINP File Management Section statement is used.

PRESTRESS

Prestress Analysis

Indicates a prestress analysis.

Format and Example	Defaults
PRESTRESS	Start run.
PRESTRESS	

Remarks

1. The following entries should be present elsewhere in the File Management Section or Bulk Data Section for a prestress analysis:

NASTDISP	Specifies an MSC Nastran displacement file to be used as input (FMS).
BULKOUT	Defines an output file to which grid-point data is written at the end of the prestress analysis (FMS).
SOLUOUT	Defines an output file to which solution data is written at the end of the prestress analysis (FMS).
NASINIT	A Bulk Data entry that controls the prestress analysis.

2. The SOLUOUT file is then used to initialize Dytran for the primary analysis (via a SOLINIT FMS statement).
3. Prestressing is described in the *Dytran User's Guide*, Chapter 9: Running the Analysis, PRESTRESS

RESTART

Restarts a Previous Run

Requests that a previous run be restarted and continued.

Format and Example	Default
RESTART	Start run.
RESTART	

Remarks

1. The RSTBEGIN File Management Section statement must be present to specify the time step from which the calculation is to be restarted.
2. The RSTFILEFile Management Section statement must be present to specify the name of the restart file to be used.
3. Restarting is described in *Dytran User's Guide*, Chapter 9: Running the Analysis, Restarting a Previous Analysis.

RSTBEGIN

Restart Time Step

Defines the time step at which a calculation is to be restarted

Format and Example	Default
`RSTBEGIN = n`	Required
`RSTBEGIN = 5000`	

Option	Meaning	Type
`n`	The number of the time step at which the analysis restarts.	I > 0

Remarks

1. A RESTART File Management Section statement must be present to indicate a restart analysis.
2. A RSTFILE File Management Section statement must be present to specify the name of the restart file to be used.
3. Restarting is described in *Dytran User's Guide*, Chapter 9: Running the Analysis, Restarting a Previous Analysis.

RSTFILE

Restart File Section

Defines the restart file to be used for restarting.

Format and Example	Default
`RSTFILE = filename`	Required
`RSTFILE = NAME.RST`	

Option	Meaning	Type
`filename`	The filename to be used for restarting. The file must exist in your runtime directory.	C

Remarks

1. A RESTART File Management Section statement must be present to indicate a Restart analysis.
2. A RSTBEGIN File Management Section statement must be present to specify the time step at which the calculations are to be restarted.
3. Restarting is described in *Dytran User's Guide*, Chapter 9: Running the Analysis, Restarting a Previous Analysis.

SAVE

Interval Between Saving an Output File

Defines how often the file is written before it is closed and saved.

Format and Example	Default
SAVE (logical_file) = n	10
SAVE (OUTPUT1) = 6	

Option	Meaning	Type
`logical_file`	The logical name of the file.	C
`n`	The number of times an output file is written before it is closed and saved. (See Remark 3.)	I

Remarks

1. When the file is written the specified number of times, it is closed, saved, and subsequent results are stored in a new file.
2. Results are available for postprocessing when the file has been closed and saved. If the `SAVE` statement is set to 1, results are stored in individual files and can be postprocessed immediately.
3. If value of n is negative for a RESTART request, the file is overwritten for every restart save. If the n value is positive, a new file is created for every restart save request.

SOLINIT

Specify an Initial Solution File from Prestress Analysis

Specifies a solution file used as input for a transient analysis of a prestressed structure.

Format and Example	Default
`SOLINIT = filename`	Required
`SOLINIT = DYTRAN.SOL`	

Option	Meaning	Type
`filename`	The filename to be used	C

Remarks

1. The `SOLINIT` File Management Section statement causes Dytran to initialize the structural part of the transient problem from a previous prestress analysis.
2. See also the BULKOUT, NASTDISP, PRESTRESS and SOLUOUT FMS statements, and the NASINIT Bulk Data entry, for performing the prestress analysis.
3. The solution file should correspond to the filename used to write out the solution data at the end of the prestress analysis (see the SOLUOUT File Management Section statement).
4. See `PARAM, INITFILE` for an overview of the different initialization methods and information on the element types for which prestressing is available.

SOLUOUT — Specifies the Output Solution File from a Prestress Analysis

Specifies an output file to which the solution data is written at the end of a prestress analysis.

Format and Example	Default
`SOLUOUT = filename`	Required
`SOLUOUT = DYTRAN.SOL`	

Option	Meaning	Type
`filename`	The filename to be used	C

Remarks

1. The solution file is a binary file. It contains all necessary data of the solution at the end of an Dytran prestress analysis.
2. See also the BULKOUT and PRESTRESS File Management Section statements, and the NASINIT Bulk Data entry.
3. The solution output file should be the same file as used for initializing the primary analysis (see the SOLINIT File Management Section statement).
4. See `PARAM,INITFILE` for an overview of the different initialization methods and information on the element types for which prestressing is available.

START

Primary Analysis

Indicates the primary analysis

Format and Example	Default
START	Primary analysis.
START	

Remarks

1. Since the default is a start analysis, this statement can be omitted.
2. See also the PRESTRESS and RESTART File Management Section statements.
3. This entry can be accompanied by using either of the following File Management Section statements:

SOLINIT	The analysis is to be initialized from a previous Dytran prestress analysis.
NASTINP	The analysis is to be initialized from a previous MSC Nastran analysis.
EULINIT	The analysis is to be initialized from a previous Dytran euler analysis.

TYPE

Type of Output File

Defines the type of an output file

Format and Example	Default
`TYPE (logical_file) = type`	`ARCHIVE`
`TYPE (OUTPUT1) = TIMEHIS`	

Option	Meaning		Type
`logical_file`	The logical filename to which the command refers.		C
`type`	The format of the file:		C
	`ARCHIVE`:	Archive file for storing results at a particular time step.	
	`TIMEHIS`:	Time-history file for storing results for particular entities at particular times during the analysis.	
	`RESTART`:	Restart file used to restart the calculation.	
	`STEPSUM`:	One-line time step summary.	
	`MATSUM`:	A material summary at a particular time step.	
	`MRSUM`:	A rigid body summary at a particular time step.	
	`EBDSUM`:	An Eulerian boundary summary at a particular time step.	

Remarks

1. Archive files are normally used to store results at one or more time steps during the analysis. Archive files are used in postprocessing to produce deformed shapes, contour plots, and vector plots. Archive files contain the model geometry and results.
2. Time-history files are normally used to store results for particular grid points and elements and are used to produce time-history plots. Only the results are stored.
3. Restart files are used to restart the calculation.
4. The summaries `STEPSUM`, `MATSUM`, `MRSUM`, and `EBDSUM` are always printed on standard output, irrespective of the value of `logical_file`.
5. The default output frequency is for serial and SMP runs is every 50 cycles for MATSUM, MRSUM, and EBDSUM and every cycle for STEPSUM. For DMP runs, the default is 1000 cycles for MATSUM, MRSUM, and EBDSUM and every 100 cycles for STEPSUM.

3 Executive Control Statements

- Overview 28
- Executive Control Summary 28
- Executive Control Descriptions 28
- CEND 29
- MEMORY-SIZE 30
- TIME 31

Overview

Executive Control is not used extensively by Dytran since, unlike MSC Nastran, it does not contain an Executive System, and DMAP is not available. It is retained for compatibility with MSC Nastran.

The Executive Control Section immediately follows the FMS and is terminated by a CEND statement. The Executive Control statements can appear in any order within the Executive Control Section.

Executive Control Summary

Currently, three Executive Control statements are available:

CEND	Marks the end of the Executive Control Section.
MEMORY-SIZE	Defines the size of the integer and float core memory. The memory requested is dynamically allocated
TIME	CPU time limit for the analysis.

Executive Control Descriptions

The format of the Executive Control statements is free field. In presenting the general formats for each statement, the following conventions are used:

- Uppercase letters should be typed as shown.
- Lowercase letters indicate that a value or option can be substituted.
- Brackets [ ] give a choice of different options.

The default value is used if the statement is not present. Where you can supply an option, the type heading indicates the type of data you must supply. This can be I (Integer), R (Real), or C (Character). A restriction on the range of the option may also be included. For example, I > 0 indicates that you must supply an integer that is greater than zero, while 0 < R < 1. indicates that you must supply a real number greater than zero and less than one.

CEND — Terminates the Executive Control Section

Marks the end of the Executive Control Section and the beginning of the Case Control Section.

Format and Example

```
CEND
```

Remark

If there are no FMS or Executive Control statements, the input file can start directly with the Case Control Section.

MEMORY-SIZE

Definition of Memory Usage

The MEMORY-SIZE statement defines the size of the integer and float core memory Dytran uses. The memory requested is dynamically allocated.

Format and Example	Default
MEMORY-SIZE = Value1, Value2	Default
MEMORY-SIZE = 250000,DEFAULT	

Option	Meaning	Type
Value1	Defines the size of the integer memory in words	I > 0
Value2	Defines the size of the float memory in words	I > 0

Remarks

1. Both values are required data. If you wish to use the default memory size for any of the values, you can use the word DEFAULT for that specific value entry. The entry is case insensitive.
2. Dytran defines the memory size according to the following rules.
   a. The user-specified definition by the MEMORY-SIZE entry always prevails.
   b. If the MEMORY-SIZE entry is not used, the default memory size depends on the setting when the analysis was submitted. On UNIX platforms, the run script takes the "size" entry that defines the memory size. The choices are small (default), medium, and large. On Windows platforms, you can define the memory size from Dytran Explorer.
   c. If you do not specify anything, the predefined default (small) applies.
3. Due to the implementation, it may still be possible that your analysis data does not fit in the requested memory. You can then alter the definition in the input file, or increase the size using Dytran Explorer. If you need an estimate of the size the analysis approximately needs, you can look at the memory summary at the end of the output file. Please note the memory sizes mentioned are only an indication, as the summary is written when the analysis stopped. If the analysis completed successfully, the core memory size is accurate.
4. There is no entry to define the character memory size. Dytran does not use any core memory character data.

TIME

Selects the Maximum CPU Time

The TIME statement is used to set the CPU time of an Dytran analysis.

Format and Example	Default
TIME = time	1 minute
TIME = 1.5	

Option	Meaning	Type
time	The maximum CPU time for the analysis in minutes.	R > 0.0

Remarks

1. When the CPU time specified on the TIME statement is used, the analysis terminates. The analysis may be continued by performing a restart, if a restart file is requested at the end of the analysis.
2. It is not possible to specify a maximum I/O time. I/O time is normally insignificant compared to the CPU time for a Dytran analysis.
3. The time is specified in minutes. Thus, 1.5 is equivalent to 90 seconds, and 480 gives 8 hours.
4. It is advised to use the TIME statement to control CPU time, rather than specifying a time limit for the batch queue or the job. If you do give a job or batch queue limit, make sure it is significantly longer than specified on the TIME statement to ensure that Dytran terminates normally and does not corrupt the files.

TIME

Selects the Maximum CPU Time

4 Case Control Commands

- Overview 36
- Case Control Summary 36
- Case Control Descriptions 38
- ACC 39
- CHECK 40
- CMARKOUT 41
- CMARKS 42
- COG 43
- CONTOUT 44
- CONTS 45
- CORDDEF 46
- CPLSBOUT 47
- CPLSOUT 48
- CPLSUBS 49
- CPLSURFS 50
- CSECS 51
- CSOUT 52
- EBDOUT 53
- EBDS 54

- ELEMENTS 56
- ELEXOUT 57
- ELOUT 58
- ENDSTEP 59
- ENDTIME 60
- GBAGOUT 61
- GBAGS 62
- GPEXOUT 63
- GPOUT 64
- GRIDS 65
- HIC 66
- INCLUDE 68
- MATOUT 71
- MATS 72
- PARAM 73
- RBOUT 74
- RELOUT 75
- RELS 76
- RIGIDS 77
- SET 78
- SETC 80
- SPC 82
- STEPS 83
- SUBSOUT 84

- SUBSURFS 85
- SURFACES 86
- SURFOUT 87
- TIC 88
- TIMES 89
- TITLE 90
- TLOAD 91
- WALLOUT 92
- WALLS 93

Overview

The Case Control Section of the input file controls the analysis, makes selections from the Bulk Data Section, and determines what results are output and how often. Case Control immediately follows the CEND statement, marking the end of the Executive Control Section, and is terminated by a BEGIN BULK entry or, in the case of a restart, by an ENDDATA entry. The Case Control commands can be in any order within the section. A summary of the commands available is given below.

Case Control Summary

The following Case Control commands are available:

Analysis Control

CHECK	Data check
ENDSTEP	Termination step for the analysis
ENDTIME	Termination time for the analysis

Data Selection

SPC	Selects single-point constraints
TLOAD	Selects transient loading
TIC	Selects transient initial conditions

Output Control

CORDDEF	Defines the moving rectangular coordinate system for deformation output
SET	Defines lists of entity numbers for use in output requests
SETC	Defines lists of names for use in output requests
TITLE	Defines the title of the analysis

Output Selection – Entity Specification

CONTS	Defines the contact surfaces for which results are to be written to a file
CPLSURFS	Defines the coupling surfaces for which results are to be written to a file
CSECS	Defines the cross sections for which results are to be written to a file
EBDS	Defines the Eulerian boundary conditions for which results are written to a file
ELEMENTS	Defines the elements for which results are to be written to a file

GBAGS	Defines the gas bags for which results are to be written to a file
GRIDS	Defines the grid points for which results are to be written to a file
MATS	Defines the materials for which results are to be written to a file
RELS	Defines the rigid ellipsoids for which results are to be written to a file
RIGIDS	Defines the rigid surfaces or MATRIG for which results are to be written to a file
SUBSURFS	Defines the subsurfaces for which results are to be written to a file
SURFACES	Defines the surfaces for which results are to be written to a file
WALLS	Defines the walls for which results are to be written to a file

Output Selection – Variable Specification

CONTOUT	Defines the contact surface data that is written to a file
CPLSOUT	Defines the coupling-surface data that is written to a file
CSOUT	Defines the cross-section data that is written to a file
EBDOUT	Defines the Eulerian boundary data that is written to a file
ELOUT	Defines the element data that is written to a file.
GBAGOUT	Defines the gas-bag data that is written to a file
GPOUT	Defines the grid-point data that is written to a file
MATOUT	Defines the material data that is written to a file
RBOUT	Defines the rigid surface, `MATRIG` or `RBE2-FULLRIG` data that is written to a file
RELOUT	Defines the rigid-ellipsoid data that is written to a file
SUBSOUT	Defines the subsurface data that is written to a file
SURFOUT	Defines the surface data that is written to a file
WALLOUT	Defines the wall data that is written to a file

Output Frequency

TIMES	Lists the times at which output is required
STEPS	Lists the time steps at which output is required

User-Defined Output

GPEXOUT	Indicates that user subroutines are used for grid point output
ELEXOUT	Indicates that user subroutines are used for element output

Input File Control

INCLUDE	Switches data input to another file

Miscellaneous

PARAM	Specifies parameters
ACC	Creates time history output of an accelerometer in local coordinate system
COG	Calculates the center of gravity for a group of elements
HIC	Calculates Head Injury Criteria (HIC) value

Case Control Descriptions

The format of the Case Control commands is free field. In presenting the general formats for each statement, the following conventions are used:

- Uppercase letters should be typed as shown.
- Lowercase letters indicate that a value or option must be substituted.
- Brackets [ ] give a choice of different options.

The default value is used if the command is not present. Where you need to supply an option, the type heading indicates the type of data you must supply. This can be `I` (Integer), `R` (Real), or `C` (Character). A restriction on the range of the option may also be included. For example, I > 0 indicates that you must supply an integer greater than zero; 0. < R < 1. indicates that you must supply a real number greater than zero and less than one.

ACC

Accelerometer Output

Creates time history output of an accelerometer in local coordinate system.

Format and Example	Default
`ACC, NAME, CID, SID, SAMPLE`	All are required.
`ACC, HEADCOGS, 3, 10, 1.E-4`	

Option	Meaning	Type
`NAME`	Unique `ACC` name	C
`CID`	Local coordinate system defined in the `BULK DATA` section.	I > 0
`SID`	Unique `SET` number	I > 0
`SAMPLE`	Dytran sampling rate. This value determines the time step when the measurements will be taken and stored in the timehistory file.	R > 0.0

Remarks:

1. For each ACC output request, a file will be generated according to: `{JOBNAME}_{NAME}_XX.THS`.
2. The set `SID` referenced must contain 1 grid point ID, which must be the same node that defines the origin of the (moving) coordinate system referenced with `CID`.
3. The ACC output request automatically stores the following variables in the timehistory file:
   a. `XPOS`, `YPOS`, and `ZPOS`: Position of accelerometer in global coordinate system.
   b. `XVEL`, `YVEL`, and `ZVEL`: The velocity of the accelerometer in coordinates of the local coordinate system.
   c. `XACC`, `YACC`, and `ZACC`: The acceleration of the accelerometer in coordinates of the local coordinate system.
   d. `RVEL` and `RACC`: The absolute velocity and absolute acceleration of the accelerometer.
4. If any `BODYFOR` boundary condition is defined for the grid point defined in the `SET`, it is subtracted from the global acceleration of the gird point. Any other acceleration fields like `GRAV`, `RFORCE`, or `ATBACC` are not subtracted from the measured grid point acceleration.

CHECK

Data Check

Selects the data checking option.

Format and Example	Default
`CHECK = [YES, NO]`	See Remark 2.
`CHECK = YES`	

Option	Meaning	Type
`YES`	A data check is performed. The analysis runs for two time steps.	C
`NO`	The analysis is run after the data is read in and checked.	C

Remarks

1. The data check option performs the following:
   a. Reads the input data.
   b. Checks for errors.
   c. Produces printed output.
   d. Runs two time steps.
   e. Writes the model data to the output files.
2. The default is `YES` for a new analysis and `NO` for a restart analysis.

CMARKOUT

Marker Data to be Output

Indicates the marker results to be written to an output file.

Format and Example	Default
`CMARKOUT (logical_file) = var1, var2, var3, ...`	No data is written.
`CMARKOUT (OUTPUT1) = PRESURE,ZVEL`	

Option	Contents	Type
`logical_file`	The logical name of the file to which the element output is written. See *Dytran's User's Guide*, Chapter 9: Running the Analysis, Outputting Results	C
var_i	Variable name to be output. See *Dytran's User's Guide*, Chapter 9: Running the Analysis, Outputting Results	C

Remarks

1. The elements for which data is written are specified using the `CMARKERS` command. The element results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of output is controlled using the TIMES and STEPS commands.
3. For a description of how to output the results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Continuation lines are not allowed when using the `CMARKOUT` command. If the `CMARKOUT` command exceeds 80 characters, a second `CMARKOUT` command (with the same `logical_file` name) can be used as follows:

```
CMARKOUT (logical_file) = var 1, var 2
CMARKOUT (logical_file) = var 3
```

CMARKS

CMARKS to be Output

Defines the CMARKS for which results are to be output to a file.

Format and Example	Default
`CMARKS (logical_file) = n`	No CMARK data is written.
`CMARKS (TH3) = 10`	

Option	Contents	Type
`logical_file`	The logical name of the file to which the element output is written.	C
n_i	Number of a SET command. Only the data for elements that appear in the set are output.	1 > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide,* Chapter 9: Running the Analysis, Outputting Results.
2. The element results written are specified using the CMARKOUT command. The element results that can be output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results. The frequency of output is controlled by using the TIMES and STEPS commands.

COG

Center of Gravity Calculation

Calculate the center of gravity for a group of elements and print the results in a time history file.

Format and Example	Default
`COG NAME TARGET SID SAMPLE`	No `COG` data is written.
`COG SEAT ... 10 1.E-4`	

Option	Contents	Type
`NAME`	Unique `COG` name	C
`TARGET`	Target type of `COG`. The ID number of the target will be defined in `SID` entry. `ELEM`: Element	C
`SID`	Unique `SET` number	I > 0
`SAMPLE`	Dytran sampling rate. This value determines the time step when center of gravity will be calculated and stored in the timehistory file.	R>0.0

Remarks

1. For each `COG` output request, a file is generated according to: `{JOB-NAME}_{NAME}_XX.THS`
2. The elements referenced in the `SET` entry can be a combination of elastic, plastic and rigid elements. However, `CONM2` definitions are not included in this calculation of the center of gravity.

CONTOUT

Contact Surface Data to be Output

Indicates the contact surface results that are to be written to an output file.

Format and Example	Default
CONTOUT (logical_file) = var1, var2, var3...	No data is written.
CONTOUT (OUTPUT1) = XFORCE, FMAGN	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the contact surface output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Outputting Results	C

Remarks

1. The contact surfaces for which data is written are specified using the `CONTS` command. The contact-surface results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled using theTIMES or STEPS command.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Continuation lines are not allowed when using the `CONTOUT` command. If the `CONTOUT` command exceeds 80 characters, a second `CONTOUT` command (with the same logical_file name) can be used as follows:

   ```
   CONTOUT (logical_file) = var1, var2
   CONTOUT (logical_file) = var3
   ```

5. For a time-history file, the following entities are written to the file together with the corresponding results:

   Primary-secondary Contact:

   C < Contact Surface ID > M:Forces/accelerations on/of the primary contact surface.
   C < Contact Surface ID > S:Forces/accelerations on/of the secondary contact surface.
   C < Contact Surface ID > T:Difference between the forces/accelerations on/of the primary and secondary contact surfaces of the contact set.

   Single-Surface Contact:

   C < Contact Surface ID > T:Forces/accelerations on/of the single surface.

   For an archive file, the combined entity C < Contact Surface ID > T is not written.

CONTS

Contact Surfaces to be Stored

Defines the contact surfaces for which results are to be output to a file

Format and Example	Default
CONTS (logical_file) = n	No contact-surface data is output.
CONTS(THS) = 14	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the contact-surface output is written.	C
`n`	Number of a SET command. Only data for contact surfaces that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The results are specified using the CONTOUT command. The contact surface results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results. The frequency of the output is controlled using the TIMES or STEPS command.
3. An archive file can contain only one contact surface definition. You can, of course, define multiple contact output definitions.
4. Contact-surface data can only be written to time-history files. (See the TYPE FMS statement.)
5. For a time-history file, the following entities are written to the file together with the corresponding results:

   Primary-secondary Contact:

   C < Contact Surface ID > M:Forces/accelerations on/of the primary contact surface.

   C < Contact Surface ID > S:Forces/accelerations on/of the secondary contact surface.

   C < Contact Surface ID > T:Difference between the forces/accelerations on/of the primary and secondary contact surfaces of the contact set.

   Single-Surface Contact:

   C < Contact Surface ID > T:Forces/accelerations on/of the single surface.

   For an archive file, the combined entity C < Contact Surface ID > T is not written.

CORDDEF

Coordinate System for Deformation Output

Defines the moving rectangular coordinate system in which the deformations are written to the archive files.

The CORDDEF entry can be added to any output request of TYPE = ARCHIVE or TIMEHIS. The grid point locations written to the archive file are the locations in the coordinate system referenced by the CORDDEF entry.

The option is particularly useful when studying the motion of a structure in a moving coordinate system

Format and Example	Default
CORDDEF(logical_file) = n	Basic system
CORDDEF(MYFILE) = 19	

Option	Meaning	Type
logical_file	The logical name of the file to which output is written.	C
n	Number of a CORDxR entry	I ≥ 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. Note that this entry is applicable only to output requests with TYPE=ARCHIVE or TIMEHIS.
3. CORDDEF updates only position for postprocessing (not XPOS, YPOS, and ZPOS), translational velocity (XVEL, YVEL, and ZVEL) and translational acceleration (XACC, YACC, and ZACC) in Archive file results. In addition, only direction from the moving coordinate of values are updated.

CPLSBOUT

Coupling Subsurface Data to be Output

Defines the coupling subsurface results to be written to an output file.

Format and Example	Default
CPLSBOUT (logical_file) = var1,var2,var3...	No data is written
CPLSBOUT (SRF_1) = XFORCE, YFORCE, ZFORCE, RFORCE	

Option	Meaning	Type
`logical_file`	The logical file name of the file to which coupling subsurface output is written	C
`var1`	Variable name to be output. See *Dytran User's Guide*,Outputting Results	C

Remarks

1. The coupling subsurfaces for which output is written must be specified using the CPLSUBS command. The coupling subsurface results available for output are defined in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled by the TIMESor the STEPS command.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.

CPLSOUT

Coupling Surface Data to be Output

Defines the coupling surface results to be written to an output file.

Format and Example	Default
CPLSOUT (logical_file) = var1,var2,var3...	No data is written
CPLSOUT (SRF_1) = PRESSURE, CLUMP, FMAT	

Option	Meaning	Type
`logical_file`	The logical file name of the file to which coupling-surface output is written	C
`var1`	Variable name to be output. See *Dytran User's Guide*,Outputting Results	C

Remarks

1. The coupling surfaces for which output is written must be specified using the CPLSURFS command. The coupling-surface results available for output are defined in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled by the TIMESor the STEPS command.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Continuation lines are not allowed when using the `CPLSOUT` command. When the command line exceeds 80 characters, a second `CPLSOUT` command (with the same logical file name) can be used as follows:

```
CPLSOUT(SRF_1) = vanr, var2
CPLSOUT (SRF_1) = var3
```

CPLSUBS

Coupling Subsurfaces to be Output

Defines the coupling subsurfaces for which results are to be output to a file. The subsurfaces have to be part of a coupling surface.

Format and Example	Default
`CPLSUBS (logical_file) = n`	No coupling subsurface is output
`CPLSUBS (SRF_1) = 44`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the coupling-subsurface output is written.	C
`n`	Number of a SET command. Only data for coupling subsurfaces that appear in the set are output.	I ≥ 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results
2. The results written are specified using the CPLSBOUT command. The coupling subsurface results available for output are described in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results
3. The frequency of output is controlled by the TIMES or STEPS command.
4. Only time history output is supported.

CPLSURFS

Coupling Surfaces to be Output

Defines the coupling surfaces for which results are to be output to a file.

Format and Example	Default
`CPLSURFS (logical_file) = n`	No coupling surface is output
`CPLSURFS (SRF_1) = 44`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the coupling-surface output is written.	C
`n`	Number of a SET command. Only data for coupling surfaces that appear in the set are output.	I ≥ 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results
2. The results written are specified using the CPLSOUT command. The coupling surface results available for output are described in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results
3. The frequency of output is controlled by the TIMES or STEPS command.

CSECS

Cross Sections to be Output

Defines the cross sections for which results are to be output to a file

Format and Example	Default
CSECS (logical_file) = n	No cross section is output.
CSECS (SEC001) = 17	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the cross-section output is written.	C
`n`	Number of a SET command. Only data for cross sections that appear in the set are output	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The results written are specified using theCSOUT command. The cross section results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results. The frequency of output is controlled using the TIMES or STEPS command.
3. Cross-section data can only be written to time-history files. (See theTYPE FMS statement.)

CSOUT

Cross-section Data to be Output

Indicates the cross-section results to be written to an output file.

Format and Example	Default
`CSOUT (logical_file) = var1, var2, var3,...`	No data is written.
`CSOUT (SEC001) = XFORCE, FMAGN`	

Option	Meaning	Type
`logical_file`	The logical file name of the file to which the cross-section output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.	C

Remarks

1. The cross sections for which output is written are specified using the CSECS command. The cross section results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results
2. The frequency of the output is controlled using the TIMES or the STEPS command.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis,Outputting Results
4. Continuation lines are not allowed when using the `CSOUT` command. If the command exceeds 80 characters, a second `CSOUT` command (with the same logical filename) can be used as follows:

   ```
   CSOUT(SEC001) = var1, var2
   CSOUT (SEC001) = var3
   ```

5. Cross-section data can only be written to time-history files. (See the TYPE FMS statement.)

EBDOUT

Eulerian Boundary Data to be Output

Indicates the Eulerian boundary results to be written to an output file.

Format and Example	Default
`EBDOUT (logical_file) = var1, var2`	No data is written.
`EBDOUT (OUTPUT1) = MFL, XMOM`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the Eulerian boundary output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.	C

Remarks

1. The Eulerian boundaries for which data is written are specified using the EBDS command. The Eulerian boundary results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Eulerian boundary data can only be written to time-history files.
5. Continuation lines are not allowed when using the `EBDOUT` command. If the `EBDOUT` exceeds 80 characters, a second `EBDOUT` (with the same logical_file name) can be used as follows:

```
EBDOUT (logical_file) = var 1, var 2
EBDOUT (logical_file) = var 3
```

EBDS

Eulerian Boundaries to be Output

Defines the Eulerian boundaries for which results are to be output to a file.

Format and Example	Default
`EBDS (logical_file) = n`	No Eulerian boundary output.
`EBDS (EBD14) = 14`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the Eulerian boundary output is written.	C
`n`	Number of a SETC command. Only data for Eulerian boundaries that appear in the set are output.	I > 0

Remarks

1. The Eulerian boundary results to be written are specified using the `EBDOUT` command. The Eulerian boundary results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The SETC command has to refer to a series of flow boundary names. These names are given by "**FLOW** boundary bulk card name(character)" + "FID(integer)". For example **FLOW,12** will give as boundary name **FLOW12**. Since Dytran explorer can only read in flow boundary names that are 8 characters or less, a flow boundary name is shortened to 8 characters if it is too long. For example using

   ```
   FLOWDIR     126     MMHYDRO    3     NEGZ     +
   ```

   would give as boundary name FLOWDIR**126**. This is shortened to **FLOWD**126. Similarly FLOWDIR12345 is shortened to **FLO**12345.

   The flow boundary names are printed in the summary of the boundary conditions in the **OUT** file:

   An example using FLOW, is as follows

   ```
   FLOW,100,5,PRESSURE,1.e+3
   ```

   gives in the **OUT** file.

   PRESCRIBED FLOW BOUNDARY - FLOW 100

   FLOW BOUNDARY ALLOWS FOR BOTH IN AND OUTFLOW

   PRESSURE - WILL BE 1.000E+03

   X-VELOCITY - WILL BE TAKEN TO BE THAT IN THE CELL ADJACENT TO THE BOUNDARY

   Y-VELOCITY - WILL BE TAKEN TO BE THAT IN THE CELL ADJACENT TO THE BOUNDARY

Z-VELOCITY - WILL BE TAKEN TO BE THAT IN THE CELL ADJACENT TO THE BOUNDARY

The flow boundary name is the value of "PRESCRIBED FLOW BOUNDARY". Therefore on the SETC entry the name FLOW100 has to be used.

An example with FLOWDIR is as follows:

```
FLOWDIR     126     MMHYDRO     3     NEGZ           +
+           FLOW      OUT
```

gives:

PRESCRIBED FLOW BOUNDARY - **FLOWD** 126

FLOW BOUNDARY ALLOWS FOR BOTH IN AND OUTFLOW

PRESSURE - WILL BE TAKEN TO BE THAT IN THE CELL ADJACENT TO THE BOUNDARY

X-VELOCITY - WILL BE TAKEN TO BE THAT IN THE CELL ADJACENT TO THE BOUNDARY

Y-VELOCITY - WILL BE TAKEN TO BE THAT IN THE CELL ADJACENT TO THE BOUNDARY

Z-VELOCITY - WILL BE TAKEN TO BE THAT IN THE CELL ADJACENT TO THE BOUNDARY

The flow boundary name to be used on the SETC entry is **FLOWD**126.

To request results for all flow boundary **ALLEBDS** can be used:

```
EBDS (EDBS) = 2
SETC 2 = ALLEBDS
```

3. The frequency of the output is controlled using the TIMES and STEPS commands.
4. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
5. Eulerian boundary results can only be written to time-history files.

ELEMENTS

Elements to be Output

Defines the elements for which results are to be output to a file.

Format and Example	Default
`ELEMENTS (logical_file) = n`	No element data is written.
`ELEMENTS (TH3) = 10`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the element output is written.	C
`n`	Number of a SET command. Only data for elements that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results. The element results written are specified using the ELOUT command. The element results that can be output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of output is controlled using theTIMES and STEPS commands.

ELEXOUT

User-defined Element Output

Output element results using a user-written subroutine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE myuds SCA.MDSolver.Obj.Uds.Dytran.InitOut
```

Format and Example	Default
`ELEXOUT (output_name)=uds_service`	No user output.
`ELEXOUT(USEROUT)=myuds`	

Option	Meaning	Type
`output_name`	The name with which the subroutine is called.	C
`uds_service`	The group name used for the FMS section CONNECT SERVICE statement	C

Remarks

1. At every time or time step specified by the TIMES or STEPS command, a subroutine named ELEXOUT is called for each of the elements listed using the ELEMENTS command allowing the user to calculate specific quantities for output.
2. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual.
4. The following commands:

   ```
   ELEXOUT (USEROUT) = MYUDS
   ELEMENTS (USEROUT) = 10
   SET 10 = 101, THRU, 110
   TIMES (USEROUT) = 1.0E-3, 2.0E-3
   ```

   cause the ELEXOUT subroutine to be called at times 1.0E–3 and 2.0E–3 for elements 101 through 110 with the use-supplied name USEROUT.

ELOUT

Element Data to be Output

Indicates the element results to be written to an output file.

Format and Example	Default
`ELOUT (logical_file) = var1, var2, var3,...`	No data is written.
`ELOUT (OUTPUT1) = TXX, TYY, TZZ`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the element output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.	C

Remarks

1. The elements for which data is written are specified using the ELEMENTS command. The element results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results. The frequency of output is controlled using the TIMES and STEPS commands.
2. For a description of how to output the results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. Continuation lines are not allowed when using the `ELOUT` command. If the `ELOUT` command exceeds 80 characters, a second `ELOUT` command (with the same `logical_file` name) can be used as follows:

```
ELOUT (logical_file) = var 1, var 2
ELOUT (logical_file) = var 3
```

ENDSTEP

Final Time Step

Defines the time-step number at which the analysis terminates

Format and Example	Default
`ENDSTEP = n`	See Remark 4.
`ENDSTEP = 3000`	

Option	Meaning	Type
`n`	The time-step number at which the transient dynamic analysis terminates.	I ≥ 0

Remarks

1. The RESTART statement can be used to continue a previous analysis. Therefore, you do not need to set `ENDSTEP` to the final point you want to reach, but instead, to the point at which you want the analysis to stop.
2. Unless you are very sure of what the analysis will do, you should always run the analysis in stages. Then use the RESTART statement to continue the analysis after you have checked how the mesh deforms.
3. The ENDTIME command can be used to terminate the analysis based on time.
4. If ENDTIME is specified, `ENDSTEP` is set to a large value (9999999).
5. At least one of the two termination criteria must be specified, either `ENDSTEP` or ENDTIME

ENDTIME

Analysis Termination Time

Defines the termination time for the analysis.

Format and Example	Default
`ENDTIME = time`	See Remark 4.
`ENDTIME = 30.0E-3`	

Option	Meaning	Type
`time`	The time, in analysis units, at which the transient dynamic analysis terminates.	R ≥ 0

Remarks

1. The RESTART statement can be used to continue a previous analysis. Therefore, you do not need to set `ENDTIME` to the final point you want to reach, but instead, to the point at which you want the analysis to stop.
2. Unless you are very sure of what the analysis will do, you should always run the analysis in stages. Then use the RESTART statement to continue the analysis after you have checked how the mesh deforms.
3. The ENDSTEP command can be used to terminate the analysis based on the number of time steps.
4. If ENDSTEP is specified, `endtime` is set to large value (99999).
5. At least one of the two termination criteria must be specified, either `ENDTIME` or ENDSTEP.

GBAGOUT

Gas Bag Data to be Output

Indicates the gas-bag results to be written to an output file.

Format and Example	Default
`GBAGOUT (logical_file) = var1, var2,...`	Required
`GBAGOUT (OUTPUT) = PRESSURE`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the gas-bag output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, GBAGOUT — Gas Bag Results	

Remarks

1. The gas bags, for which data is written, are specified using the GBAGS command. The gas-bag results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, GBAGOUT — Gas Bag Results.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, GBAGOUT — Gas Bag Results.
4. Gas-bag data can only be written to time-history files. (See the TYPE FMS statement.)
5. Continuation lines are not allowed when using the `GBAGOUT` command. If the `GBAGOUT` command exceeds 80 characters, a second `GBAGOUT` command (with the same `logical_file` name) can be used as follows:

```
GBAGOUT (logical_file) = var 1, var 2
GBAGOUT (logical_file) = var 3
```

GBAGS

Gas Bags to be Output

Defines the gas bags for which results are to be output to a file.

Format and Example	Default
`GBAGS (logical_file) = n`	No gas bag data is output.
`GBAGS(THG) = 14`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the gas-bag output is written.	C
`n`	Number of a SET command. Only data for gas bags that appear in the set are output.	I ≥ 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The results written are specified using the `GBAGOUT` command. The gas bag results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, GBAGOUT — Gas Bag Results.
3. The frequency of output is controlled using theTIMES and STEPS commands.
4. Gas bag data can only be written to time-history files. (See theTYPE FMS statement.)

GPEXOUT

User-defined Grid Point Output

Output grid-point results using a user-written subroutine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE myuds SCA.MDSolver.Obj.Uds.Dytran.InitOut
```

Format and Example	Default
`GPEXOUT (output_name)=uds_service`	No user output.
`GPEXOUT (DYTRAN_EXT_GP)=myuds`	

Option	Meaning	Type
`output_name`	Name used when subroutine is called.	C
`uds_service`	The group name used for the FMS section CONNECT SERVICE statement	C

Remarks

1. At every time or time step specified by the TIMES or STEPS commands, a subroutine called `GEXOUT` is called for each of the grid points specified using a GRIDS command that allows you to calculate specific quantities for output.
2. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results
3. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual.
4. The following commands:

   ```
   GPEXOUT (DYTRAN_EXT_GP)= myuds
   GRIDS (DYTRAN_EXT_GP) = 3
   SET, 3, 1 THRU 35.
   STEPS (DYTRAN_EXT_GP) = 5, 10, 15
   ```

   cause the GPEXOUT user subroutine to be called at time steps 5, 10, and 15 for grid points 1 through 35 with the user-supplied name `DYTRAN_EXT_GP`.

GPOUT

Grid Point Data to be Output

Indicates the grid point results to be written to an output file.

Format and Example	Default
`GPOUT (logical_file) = var1, var2, var3,...`	No data is written.
`GPOUT (OUTPUT1) XVEL, XFORCE`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the grid-point output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results	C

Remarks

1. The grid points for which data is written are specified using theGRIDS command. The grid-point results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Continuation lines are not allowed when using the `GPOUT` command. If the `GPOUT` command exceeds 80 characters, a second `GPOUT` command (with the same `logical_file` name) can be used as follows:

```
GPOUT (logical_file) = var 1, var 2
GPOUT (logical_file) = var 3
```

GRIDS

Grid Points to be Output

Defines the grid points for which results are to be output to a file.

Format and Example	Default
`GRIDS (logical_file) = n`	No grid-point output.

Option	Meaning	Type
`logical_file`	The logical name of the file to which the grid-point output is written.	C
`n`	Number of a SET command. Only data for grid points that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The grid point results to be written are specified using the `GPOUT` command. The grid point results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. The frequency of output is controlled using theTIMES and STEPS commands.

HIC

Head Injury Criteria Calculation

Calculate Head Injury Criteria (HIC) value, its time duration and resultants from head acceleration time history curve, by means of the maximum of an integral with variable limits. Two different definitions are used in HIC calculation: unlimited time envelope and a limited time envelope.

Format and Example	Default
`HIC, NAME, TARGET, SID, HIC TYPE, SAMPLE, THSOUT`	All are required.
`HIC, ELL01, ELLIPS, 10, 0.036, 1.E-4, NO`	

Option	Meaning		Type
`NAME`	Unique HIC name		C
`TARGET`	Target type of HIC. The Name or ID number of the target will be defined in SID entry.		C
	ELLIPS	Ellipsoid	
	GRID	Grid point	
	RIGID	Rigid	
`SID`	Unique `SET` or `SETC` number		I > 0
`HIC TYPE`	The definition of HIC calculation. If a limited time envelope is chosen then this entry provides the range of time envelope.		C or R>0.0
	`UNLIMITED`	unlimited time envelope	
	`REAL VALUE`	value indicates the range of time	
`SAMPLE`	Dytran sampling rate (in seconds). This value determines the time step where the head acceleration will be stored and used for HIC calculation.		R>0.0
`THSOUT`	Option to write time history curve file.		C
	YES	Time history file will be written with HIC name	
	NO	No time history curve file	

Remarks

1. The summary of `HIC` calculation will be written to `*.OUT` file
2. If the sampling rate (`SAMPLE`) is close to the delta time step of Dytran, then a warning message is written to the `*.OUT` file of suspicious HIC values.
3. The value of the gravity used by the HIC calculations has to be set by the parameter HICGRAV. When this value is not set, a warning message is issued and the default value of 9.80665 is used.

INCLUDE

Starts Reading of a New File

Switches reading of the input data to another file. Once that file has been read, processing returns to the original file immediately after the `INCLUDE` file.

Format and Example	Default
`INCLUDE filename`	Read `.dat` file.
`INCLUDE INPUT.DAT`	

Option	Meaning	Type
`filename`	The name of the new input file to be used. The name must be appropriate to the machine on which Dytran is executing.	C

Remarks

1. The file must be present in the working area where Dytran is executing.
2. BEGIN BULK and ENDDATA may be included in an `INCLUDE` file.

MATBOUT

MATBX Data to be Output

Indicates the matbx results to be written to an output file.

Format and Example	Default
`MATBOUT (logical_file) = var1, var2,...`	No data is written.
`MATBOUT (OUTPUT1) = XMOM, YMOM`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the material output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, MATOUT — Material Results.	C

Remarks

1. The boxes for which data is written are specified using the MATBX command. The **MATBX** results that can be requested for output are listed in the *Dytran User's Guide*, Chapter 9: Running the Analysis, Time History Output for Boxes. All Eulerian element variables can be requested. In most cases, results are the mass weighted average of all element values that are inside a box. Exceptions are `MASS`, `ENER`, `XMOM`, `YMOM`, `ZMOM`. These are not averaged but simply accumulated. Material Fractions are weighted with the uncovered volume of the elements inside the box. The uncovered volume fraction `FVUNC` is weighted with total element volume.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Time History Output for Boxes.
4. MATBX data can only be written to time-history files.
5. Continuation lines are not allowed when using the `MATBOUT` command. If the `MATBOUT` command exceeds 80 characters, a second `MATBOUT` command (with the same `logical_file` name) can be used as follows:

```
MATBOUT (logical_file) = var 1, var 2
MATBOUT (logical_file) = var 3
```

MATBX

Boxes to be Output

Defines the boxes for which results are to be output to a file.

Format and Example	Default
`MATBX (logical_file) = n`	No element data is written.
`MATBX (TH3) = 10`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the **MATBX** output is written.	C
`n`	Number of a SET command. Only data for elements that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see the *Dytran User's Guide*, Chapter 9: Running the Analysis, Time History Output for Boxes.
2. The box results written are specified using the MATBOUT command. The element results that can be output are listed in the *Dytran User's Guide*, Chapter 9: Running the Analysis, Time History Output for Boxes.
3. For each box, it is determined which elements are inside the box. Elements that are only partially inside a box will also be partially be taken into account.
4. **MATBX** is only available for Eulerian elements and fully supports multiple coupling surfaces.
5. **MATBX** can be seen as a CHEXA marker. For non-orthogonal meshes, markers are not supported and **MATBX** is then a good alternative. There are no limits on the volume of the box, so the box can be chosen small such that matbx resembles a marker. Defining many **MATBX** entries will not significantly increase simulation times.
6. The frequency of output is controlled using the TIMES and STEPS commands.
7. An example is given below

```
TYPE    (ARCMATBX) = TIMEHIS
MATBOUT (ARCMATBX) = DENSITY4,DENSITY4,MASS100,MASS100,
                     XVEL,YVEL,ZVEL,FMAT4,FMAT100,PRESSURE
MATBX   (ARCMATBX) = 70
SET 70 = 50,60
STEPS   (ARCMATBX) = 0,THRU,END,BY,1
SAVE    (ARCMATBX) = 99999
BOX,50,,0.5625,-1,9.68,3.75,2,0.93
BOX,60,,0.5625,-1,4.35,3.75,2,0.93
```

MATOUT

Material Data to be Output

Indicates the material results to be written to an output file.

Format and Example	Default
`MATOUT (logical_file) = var1, var2,...`	No data is written.
`MATOUT (OUTPUT1) = XMOM, YMOM`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the material output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, MATOUT — Material Results.	C

Remarks

1. The materials for which data is written are specified using the MATS command. The material results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results,MATOUT — Material Results.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, MATOUT — Material Results .
4. Material data can only be written to time-history files.
5. Continuation lines are not allowed when using the `MATOUT` command. If the `MATOUT` command exceeds 80 characters, a second `MATOUT` command (with the same `logical_file` name) can be used as follows:

```
MATOUT (logical_file) = var 1, var 2
MATOUT (logical_file) = var 3
```

MATS

Materials to be Output

Defines the materials for which results are to be output to a file.

Format and Example	Default
`MATS (logical_file) = n`	No material output.
`MATS (MAT19) = 19`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the material output is written.	C
`n`	Number of a SET command. Only data for materials that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The material results to be written are specified using the MATOUT command. The material results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. The frequency of output is controlled using the TIMES and STEPS commands.
4. Material results can only be written to time-history files.

PARAM

Parameter Specification

Defines the values for the parameters that are used during the analysis

Format and Example	Default
`PARAM, name, value`	See Chapter 6: Parameters in this manual
`PARAM, INISTEP, 1.E-7`	

Option	Meaning	Type
`name`	Parameter name	C
`value`	Value associated with name	I, R, C

Remark

This command is normally used in the Bulk Data Section. A list of parameters that can be set, along with the parameter names and values, is given in Chapter 6: Parameters of this manual.

RBOUT

Rigid Body Data to be Output

Indicates the rigid body results to be written to an output file.

Format and Example	Default
`RBOUT (logical_file) = var1, var2`	No data is written.
`RBOUT (OUTPUT1) = XVEL, YVEL, XAVEL, YAVEL, ZAVEL`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the rigid body output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results	C

Remarks

1. The rigid bodies for which data is written are specified using the RIGIDS command. The rigid body results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Continuation lines are not allowed when using the `RBOUT` command. If the `RBOUT` command exceeds 80 characters, a second `RBOUT` (with the same `logical_file` name) can be used as follows:

```
RBOUT (logical_file) = var 1, var 2
RBOUT (logical_file) = var 3
```

RELOUT

Rigid Ellipsoid Data to be Output

Indicates the rigid ellipsoid results to be written to an output file.

Format and Example	Default
`RELOUT (logical_file) = var1, var2`	No data is written.
`RELOUT (OUTPUT1) = GEOMETRY`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the rigid ellipsoid output is written.	C
`vari`	Variable name to be output. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, RELOUT — Rigid Ellipsoid Results.	C

Remarks

1. The rigid ellipsoids for which data is written are specified using the `RELS` command. The rigid ellipsoid results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, RELOUT — Rigid Ellipsoid Results.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results, RELOUT — Rigid Ellipsoid Results.
4. The keyword `GEOMETRY` causes a mesh to be placed on the rigid ellipsoids for visualization purposes in the postprocessor. This keyword can be used only with archive files.
5. Continuation lines are not allowed when using the `RELOUT` command. If the `RELOUT` command exceeds 80 characters, a second `RELOUT` command (with the same `logical_file` name) can be used as follows:

```
RELOUT (logical_file) = var 1, var 2
RELOUT (logical_file) = var 3
```

RELS

Rigid Ellipsoids to be Output

Defines the rigid ellipsoids for which results are to be output to a file.

Format and Example	Default
`RELS (logical_file) = n`	No rigid-ellipsoid output.
`RELS (FILE_REL) = 170`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the rigid ellipsoid output is written.	C
`n`	Number of a SETC command. Only data for rigid ellipsoids that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The rigid ellipsoid results to be written are specified using the RELOUT command. The rigid-ellipsoid results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. The frequency of output is controlled using the TIMES and STEPS commands.
4. A SETC is used to enable output for rigid ellipsoids obtained from `ATB`.

RIGIDS

Rigid Bodies to be Output

Defines the rigid bodies for which results are to be output to a file.

Format and Example	Default
`RIGIDS(logical_file) = n`	No rigid-body output.
`RIGIDS (TH5Z) = 32`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the user output is written.	
`n`	Number of a SET command. Only data for rigid bodies that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The rigid-body results to be written are specified using the RBOUT command. The rigid-body results that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. The SET can refer to a `RIGID` surface (`id`), a `MATRIG` (`MR<id>`), or an `RBE2-FULLRIG` (`FR<id>`).
4. The frequency of output is controlled using the TIMES and STEPS commands.

SET

Set Definition

Defines a list of grid points, elements, etc., for which output is required.

Format and Example	Default
`SET n = i1,[, i2, i3 THRU i4 BY i5]`	Required
`SET 77 = 5`	
`SET 88 = 5, 6, 7, 8, 9, 10 THRU 55 BY 3`	
`15, 16, 77, 78, 79, 100 THRU 300 BY 2`	
`SET 99 = 1 THRU 100000`	
`SET 44 = ALLSHQUAD`	

Option	Meaning	Type
`n`	Set number.	I > 0
`i1, i2 etc.`	Element or grid-point number at which the output is requested.	I > 0
`i3 THRU i4 BY i5`	Output at numbers `i3` to `i4` (`i4 > i3`) with an increment of `i5`.	I > 0
`ALLSHQUAD`	Data is output for all entities associated with quadrilateral shell elements or grid points. (CQUAD4)	C
`ALLSHTRIA`	Data is output for all entities associated with triangular shell elements or grid points. (CTRIA3)	C
`ALLMEMTRIA`	Data is output for all entities associated with triangular membrane elements or grid points. (CTRIA3)	C
`ALLLAGSOLID`	Data is output for all entities associated with Lagrangian solid elements or grid points.	C
`ALLEULHYDRO`	Data is output for all entities associated with hydrodynamic Eulerian elements or grid points.	C
`ALLEULSTRENGTH`	Data is output for all entities associated with Eulerian elements or grid points with shear strength.	C
`ALLDUMQUAD`	Data is output for all entities associated with dummy CQUAD4 elements or grid points.	C
`ALLDUMTRIA`	Data is output for all entities associated with dummy CTRIA3 elements or grid points.	C
`ALLMULTIEULHYDRO`	Data is output for all entities associated with hydrodynamic Eulerian multimaterial elements or grid points.	C
`ALLMULTIEULSTREN`	Data is output for all entities associated with Eulerian multimaterial elements or grid points with shear strength.	C

Option	Meaning	Type
ALLELEM1D	Data is output for all entities associated with one-dimensional elements or grid points.	C
ALLELEMENTS	Data is output for all entities associated with all elements.	C
ALLGRIDPOINTS	Data is output for all entities associated with all grid points.	C
ALLCONTACTS	Data is output for all entities associated with all contacts.	C
ALLCSECS	Data is output for all entities associated with all cross sections.	C
ALLMAT	Data is output for all materials.	C
ALLEULMAT	Data is output for all Eulerian materials.	C
ALLLAGMAT	Data is output for all Lagrangian materials.	C

Remarks

1. A SET command may occupy more than one line in the input file. A comma (,) at the end of a line signifies that the next line is a continuation. Commas cannot end a set.
2. The keyword BY does not have to be used when specifying an i1 THRU i2 range since the assumed default is 1.

SETC

List of Names

Defines a list of names (character strings) that are used to specify what output is required.

Format and Example	Default
`SETC n = name1, name2, name3,name4`	Required.
`SETC 10 = HUB, RIM, FLOW200, ALE2`	

Option	Meaning	Type
`n`	`SETC` number.	I > 0
`namei`	Character string.	C

Remarks

1. A `SETC` command may occupy more than one line of the input file. A comma (,) at the end of a line signifies that the next line is a continuation. Commas cannot end a set.
2. This `SETC` may be referred to from outside the Case Control Section.
3. The length of the character string must be 16 characters or less.
4. The RELS command uses the `SETC` instead of the normal `SET1`, enabling the user to specify character strings rather than integers.

SGAUGES

Surface Gauges to be Stored

Defines the surface gauges for which results are to be output to a file.

Format and Example	Default
`SGAUGES (logical_file) = n`	No surface gauge data is output.
`SGAUGES (SG12) = 245`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the surface gauge output is written.	C
`n`	Number of a SET command. Only data for surface gauges that appear in the set are output.	I > 0

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. Surface gauge data can only be written to time-history files. (See the TYPE FMS statement).

SPC

Single Point Constraint Set Selection

Selects the single-point constraints to be used.

Format and Example	Default
`SPC = n`	No SPCs are used.
`SPC = 100`	

Option	Meaning	Type
`n`	Number of a set of `SPC`, `SPC1`, `SPC2`, and SPC3 entries to be used.	I > 0

Remark

Single point constraints are not used by Dytran unless they are selected in the Case Control Section.

STEPS

Time Steps at which Data is Written

Defines the time steps at which data is written to an output file

Format and Example	Default
`STEPS (logical_file) = i1, [i2, i3, THRU, i4, BY, i5]`	Required
`STEPS (OUTPUT1) = 0, THRU, END, BY, 100`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the user output is written.	C
`i1, i2, etc.`	Time steps at which output is required.	I
`i3,THRU, i4 BY, i5`	Time steps i3 to i4 using an increment i5 (i4 > i3).	I

Remarks

1. The keyword `END` can be used to indicate the end of the calculation.
2. The TIMES command can be used instead to control the output using the values of time.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. A list of steps should be in ascending order.

SUBSOUT

Subsurface Data to be Output

Indicates the subsurface results that are to be written to an output file.

Format and Example	Default
`SUBSOUT(logical_file)`	var1, var2, var3...
`SUBSOUT(SUBSURF)`	TEMPTURE, MSFR, PRESSURE

Option	Meaning
`logical_file`	The logical name of the file to which the subsurface output is written.
`vari`	Variable name to be output.

Remarks

1. The subsurfaces for which data is written are specified using the `SUBOUT` command. The subsurface data that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Subsurface output data can only be written to a time history files. (See theTYPE FMS statement.)

SUBSURFS

Subsurfaces to be Stored

Defines the subsurfaces for which results are to be written to a file.

Format and Example	Default
`SUBSURFS (logical_file)`	N
`SUBSURFS (SUBSURF)`	14

Option	Meaning
`logical_file`	The logical name of the file to which the subsurface output is written.
`n`	Number of a SET command. Only data for `GBAG` or `COUPLING` subsurfaces that appear in the set are output.

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The results are specified using the SUBSOUT command. The subsurf data that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. The frequency of the output is controlled using theTIMES or STEPS command.
4. Subsurface output data can only be written to a time history files. (See the TYPE FMS statement).
5. The `SUBSURFACE`s specified in the SET command need to be part of a `SURFACE` referenced by a `COUPLE` or `GBAG` entry.

SURFACES

Surfaces to be Stored

Defines the surfaces for which results are to be written to a file.

Format and Example	Default
`SURFACES(logical_file)`	N
`SURFACES(SURF_1)`	14

Option	Meaning
`logical_file`	The logical name of the file to which the surface output is written.
`n`	Number of a SET command. Only data for GBAG or `COUPLING` surfaces that appear in the set are output.

Remarks

1. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The results are specified using the SURFOUT command. The subsurf data that can be requested for output is listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
3. The frequency of the output is controlled using the TIMES or STEPS command.
4. Surface output data can only be written to a time history files. (See the TYPE FMS statement).
5. The `SURFACE`s specified in the SET command need to be referenced by a COUPLE or GBAG entry.

SURFOUT

Surface Data to be Output

Indicates the surface results that are to be written to an output file.

Format and Example	Default
`SURFOUT(logical_file)`	var1, var2, var3...
`SURFOUT(SURF_1)`	TEMPTURE, MSFR, PRESSURE

Option	Meaning
`logical_file`	The logical name of the file to which the subsurface output is written.
`vari`	Variable name to be output.

Remarks

1. The surfaces for which data is written are specified using the SURFACES command. The surface data that can be requested for output are listed in *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
2. The frequency of the output is controlled using theTIMES or STEPS command.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. Surface output data can only be written to a time history files. (See the TYPE FMS statement.)

TIC

Transient Initial Condition Selection

Selects the transient initial conditions to be used.

Format and Example	Default
`TIC = n`	No initial conditions are applied.
`TIC = 42`	

Option	Meaning	Type
`n`	Number of a set of `TIC`, `TIC1`, `TIC2`, `TICGP`, or `TICEL` to be used.	I > 0

Remark

Initial conditions are not used by Dytran unless they are selected in the Case Control Section.

TIMES

Times at which Data is Written

Defines the times at which data is to be written to an output file.

Format and Example	Default
`TIMES (logical_file) = t1, [t2, t3, THRU, t4, BY, t5]`	Required.
`TIMES (OUTPUT1) = 0.0, THRU, 5.0, BY, 0.5, 0.6, THRU, END, BY, 0.03`	
`TIMES (ARC) = 1.0E-3, 3.0E-3, 7.-3`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the user output is written.	C
`t1, t2, etc.`	Times at which output is required.	R
`t3, THRU, t4 BY, t5`	Times `t3` to `t4` using an increment `t5` (`t4 > t3`).	R

Remarks

1. The keyword `END` can be used to indicate the end of the calculation.
2. The STEPS command can be used instead to control the output using the time-step numbers.
3. For a description of how to output results, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Outputting Results.
4. A list of times should be in ascending order.

TITLE

Output Title

Defines the title for the Analysis

Format and Example	Default
`TITLE = string`	No title
`TITLE = ANALYSIS - run 13`	

Option	Meaning	Type
`string`	A string of up to 72 alphanumeric characters giving a title for the analysis.	C

Remark

The title is written to the output files for use in postprocessing.

TLOAD

Transient Load Selection

Selects the transient loading to be applied.

Format and Example	Default
`TLOAD = n`	No loads are applied.
`TLOAD = 2`	

Option	Meaning	Type
`n`	Number of a set of TLOAD1 or TLOAD2 entries.	I > 0

Remark

Loads, pressures, flow boundaries, and enforced motion are not used by Dytran unless they are selected in the Case Control Section.

WALLOUT

Wall Data to be Output

Indicates the wall results to be written to an output file.

Format and Example	Default
`WALLOUT (logical_file) = var1, var2`	No data is written.
`WALLOUT (OUTPUT1) = XFORCE, YFORCE, FMAGN, XACC, AMAGN`	

Option	Meaning	Type
`logical_file`	Number of a set of TLOAD1 or TLOAD2 entries.	C
`vari`	Variable name to be output. See Running the Analysis (Ch. 9) in the *Dytran User's Guide*, Outputting Results.	C

Remark

1. The walls for which data is written are specified using the WALLS command. The wall data that can be requested for output are listed in Running the Analysis (Ch. 9) in the *Dytran User's Guide*, Outputting Results.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. For a description of how to output results, See Running the Analysis (Ch. 9) in the *Dytran User's Guide*, Outputting Results.
4. Wall output data can only be written to a time history files. (See the TYPE FMS statement.)

WALLS

Walls to be Output

Defines the walls for which results are to be output to a file.

Format and Example	Default
`WALLS (logical_file) = n`	No wall output.
`WALLS (OUTPUT2) = 2`	

Option	Meaning	Type
`logical_file`	The logical name of the file to which the user output is written.	C
`vari`	Number of a SET command. Only data for walls that appear in the	I>0

Remark

1. For a description of how to output results, see Running the Analysis (Ch. 9) in the *Dytran User's Guide*, Outputting Results.
2. The wall results to be written are specified using the WALLOUT command.
3. The SET refers to WALL(id)
4. The frequency of the output is controlled using the TIMES or STEPS command.
5. Wall output data can only be written to a time history files. (See the TYPE FMS statement.)

5 Bulk Data Entry Descriptions

- Overview 106
- Format of Bulk Data Entries 106
- Bulk Data Summary 108
- Bulk Data Descriptions 120
- $ 121
- ACTIVE 122
- ALE 124
- ALEGRID 125
- ALEGRID1 128
- ATBACC 131
- ATBJNT 132
- ATBSEG 135
- BARLT89 137
- BARLT00 142
- BEGIN BULK 146
- BIAS 147
- BJOIN 150
- BODYFOR 154
- BODYFR1 156
- BOX 158
- BOX1 159

- CBAR 160
- CBEAM 162
- CBUSH 164
- CDAMP1 166
- CDAMP2 168
- CELAS1 170
- CELAS2 172
- CFACE 174
- CFACE1 176
- CHEXA 177
- CIFHEX 179
- CMARKB2 180
- CMARKN1 181
- COHFRIC 182
- CONM2 183
- CONTACT 184
- CONTFORC 204
- CONTINI 206
- CONTREL 208
- CORD1C 209
- CORD1R 211
- CORD1S 213
- CORD2C 215
- CORD2R 217
- CORD2S 219
- CORD3R 221
- CORD4R 223
- CORDROT 225
- COUHTR 227
- COUINFL 229
- COUOPT 231

- COUP1INT 234
- COUPLE 235
- COUPLE1 239
- COUPOR 241
- CPENTA 246
- CQUAD4 247
- CROD 249
- CSEG 250
- CSPR 251
- CTETRA 252
- CTRIA3 253
- CVISC 255
- CYLINDER 256
- DAREA 257
- DETSPH 258
- DMAT 259
- DMATEL 261
- DMATEP 263
- DMATOR 264
- DRAWBD 267
- DRAWBD1 269
- DYMAT14 271
- DYMAT24 274
- DYMAT25 276
- DYMAT26 278
- ENDDATA 282
- EOSDEF 283
- EOSEX 286
- EOSGAM 288
- EOSIG 290

- EOSJWL 296
- EOSMG 298
- EOSNA 300
- EOSPOL 302
- EOSTAIT 305
- FABRIC 307
- FAILEST 311
- FAILEX 312
- FAILEX1 313
- FAILEX2 314
- FAILJC 315
- FAILMES 317
- FAILMPS 318
- FAILPRS 319
- FAILSDT 320
- FFCONTR 321
- FLOW 323
- FLOWC 325
- FLOWCDR 328
- FLOWCSQ 331
- FLOWDEF 334
- FLOWDIR 336
- FLOWEX 338
- FLOWSQ 340
- FLOWT 343
- FLOWTSQ 346
- FLOWXDR 350
- FLOWXSQ 352
- FOAM1 354

- FOAM2 356
- FORCE 359
- FORCE1 361
- FORCE2 362
- FORCE3 363
- FORCEEX 365
- GBAG 366
- GBAGC 373
- GBAGCOU 376
- GBAGHTR 377
- GBAGINFL 379
- GBAGPOR 381
- GRAV 384
- GRDSET 385
- GRID 386
- GROFFS 387
- HGSUPPR 388
- HTRCONV 391
- HTRRAD 392
- HYDSTAT 393
- IGNORE 395
- INCLUDE 396
- INFLATR 397
- INFLATR1 399
- INFLCG 401
- INFLFRAC 403
- INFLGAS 404
- INFLHYB 406
- INFLHYB1 407

- INFLTANK 408
- INITGAS 410
- JOIN 411
- KJOIN 412
- MAT1 413
- MAT2 414
- MAT8 415
- MAT8A 416
- MATINI 421
- MATRIG 423
- MCOHE 426
- MESH 430
- MOMENT 439
- MOMENT1 440
- MOMENT2 441
- NASINIT 442
- PARAM 443
- PBAR 444
- PBCOMP 445
- PBEAM 447
- PBEAM1 449
- PBEAM1 453
- PBEAM1 455
- PBEAML 460
- PBELT 464
- PBUSH 466
- PBUSHT 467
- PCOHE 472
- PCOMP 473
- PCOMPA 475
- PDAMP 477

- PELAS1 479
- PELASEX 480
- PERMCPL 482
- PERMEAB 484
- PERMGBG 486
- PEULER 487
- PEULER1 489
- PLOAD 490
- PLOAD4 491
- PMARKER 494
- PMINC 495
- POREX 497
- PORFCPL 498
- PORFGBG 499
- PORFLCPL 500
- PORFLGBG 501
- PORFLOW 503
- PORFLOWT 505
- PORHOLE 508
- PORHYDST 509
- PORLHOLE 510
- PROD 512
- PSHELL 513
- PSHELL1 515
- PSOLID 518
- PSPR 520
- PSPR1 521

- PSPREX 522
- PVISC 523
- PVISC1 524
- PVISCEX 525
- PWELD 526
- PWELD1 529
- PWELD2 533
- RBC3 536
- RBE2 538
- RBHINGE 540
- RCONN 541
- RCONREL 544
- RELEX 545
- RELLIPS 546
- RFORCE 547
- RIGID 548
- RJCYL 550
- RJPLA 552
- RJREV 554
- RJSPH 556
- RJTRA 558
- RJUNI 560
- RUBBER1 562
- RUBBER2 564
- SECTION 566
- SET1 567
- SETC 568
- SETTING 569
- SHEETMAT 571

- SHREL 576
- SHREX 577
- SHRLVE 578
- SHRPOL 580
- SPC 581
- SPC1 582
- SPC2 583
- SPC3 585
- SPHERE 587
- SUBSURF 588
- SURFACE 590
- TABFILE 592
- TABLED1 593
- TABLEEX 595
- TIC 596
- TIC1 597
- TIC2 598
- TIC3 599
- TICEEX 600
- TICEL 601
- TICEUL 602
- TICGEX 604
- TICGP 605
- TICVAL 606
- TLOAD1 608
- TLOAD2 610
- VISCDMP 612
- WALL 615
- WALLDIR 617

- WALLET 618
- YLDEX 619
- YLDHY 620
- YLDJC 621
- YLDMC 623
- YLDMSS 624
- YLDPOL 626
- YLDRPL 627
- YLDSG 628
- YLDTM 630
- YLDVM 632
- YLDZA 635

Overview

The Bulk Data Section of the input file contains all the data to fully describe the analysis model, including the geometry, topology, constraints, and loading. This section must begin with a BEGIN BULK entry. Thereafter, entries can appear in any order except that continuation lines must follow the entry from which they are referenced. Entries can be numbered in any manner that is convenient. Gaps in the numbering are allowed. The input file must finish with an ENDDATA entry.

Many of the entries are the same as those used for MSC Nastran. However, sometimes not all the fields are used for Dytran. If data occurs in the unused fields, a User Warning Message is issued and the excess data is ignored, see Chapter 1: Introduction, Similarity with MSC Nastran in this manual. Similarly, any MSC Nastran entry that is not used by Dytran is ignored.

Format of Bulk Data Entries

A Bulk Data entry consists of one or more lines in the input file. The first line starts with a mnemonic that identifies the entry and is called the parent entry. Any other lines are called continuations. Each line can be in free or fixed format. In free format, the fields can appear anywhere on the line and are separated by commas. With fixed format, each field must be located in a set part of the line. There are two types of fixed format: small and large. Small format consists of ten fields, each of which has eight characters. The entire entry is defined on a single line of the input file. Large format splits the entry so that it occupies two lines of the input file. Each line consists of one field of eight characters, four fields of sixteen characters, and one of eight characters. Small- and large-format entries must be in fixed format, that is, the data must be entirely within the columns that make up the field.

Free and fixed field lines can be freely mixed in the input file so, for example, a fixed format entry can have a free format continuation, or vice versa.

The first field of each Bulk Data entry contains a mnemonic that identifies the type of entry. Fields 2 through 9 contain data, while field 10 is used for a continuation identifier or for user comment if there are no continuation lines. The mnemonic must start in column one of the first field.

Fields 2 through 9 are for data items. The only limitations on data items are that they cannot have embedded blanks and must be of the proper type; that is, blank, Integer, Real, or Character. A blank is interpreted as a real default value. Real numbers may be encoded in various ways. For example, the real number 7.0 may be encoded as 7.0, .7E1, 0.7+1, 70.–1, 7+0, 7, etc. Character data values consist of one to eight alphanumeric characters, the first of which must be alphabetic.

In the case of continuation lines, the first character of Field 10 must contain a +, and the first character of Field 1 on the next line must contain a +.

Free Field Format

With free field input, the position of the data items on the line is irrelevant. The mnemonic and data items must be separated by commas. For example:

```
GRID, 7, 0, 0.0, 1.0, 3.7569
```

Free-field entries must start in column one; data fields can consist of any number of characters as long as the whole entry fits on one line of 80 characters.

A field may be left blank by entering two commas with or without spaces between them:

```
GRID, 7, , 0.0, 1.0, 3.7569
```

Only those fields containing data need be entered. All the extra fields are given their default values. In the example above, only six fields have been entered, so the last four are set to the default.

Small Field, Fixed Format Entry

1	2	3	4	5	6	7	8	9	10
1a	2	3	4	5	6	7	8	9	10a
8	8	8	8	8	8	8	8	8	8

The small-field, fixed-format entry consists of a single line in the input file containing 80 characters and comprising 10 fields, each of which has eight characters. The data in each field must lie completely within the designated columns.

Large Field, Fixed Format Entry

The small-field format should be adequate for most applications. Occasionally, however, the input is generated by another computer program or is available in a form where a wider field is desirable. For these cases, the larger field format with a 16-character data field is provided. Two lines of the input file are used as indicated:

1a	2	3	4	5	6a
8	16	16	16	16	8

1b	2	3	4	5	6b
8	16	16	16	16	8

The large field format is denoted by placing the symbol * after the mnemonic in Field 1a and a + as the first character of Field 10a. The second line contains the symbol * in column one. The second line may, in turn, be used to point to a large or small field continuation line, depending on whether the continuation line contains the symbol * (for a large field) or the symbol + (for a small field) in column one. The use of multiple and large field lines is illustrated in the following examples:

Small Field Entry with Small-Field Continuation

1	2	3	4	5	6	7	8	9	10
TYPE									+
+									

Large Field Entry

TYPE*									+
+									

Large Field Entry with Large Field Continuation

TYPE*					+
*					+
*					+
*					

Large Field Entry Followed by a Small Field Continuation and a Large Field Continuation

TYPE*											+
*											+
+											+
*											+
+											

Small-Field Entry with Large Field Continuation

TYPE									+
+									+
*									

Bulk Data Summary

This section contains a summary of all the Bulk Data entries under the following subsections:

- Geometry
- Lagrangian and Eulerian Elements
- Constitutive Models
- Rigid Bodies
- Lagrangian Constraints
- Lagrangian Loading
- Eulerian Loading and Constraints
- Euler/Lagrange Coupling
- Miscellaneous

Geometry

Grid Points

GRID	Grid-point location, coordinate system selection.
GRDSET	Default options for GRID entries.

GROFFS	Grid-point offset in the local coordinate system.
CONM2	Concentrated grid-point mass and/or inertia.

Coordinate Systems

CORD1R,CORD2R	Rectangular coordinate system definition
CORD1C, CORD2C	Cylindrical coordinate system definition
CORD1S, CORD2S	Spherical coordinate system definition
CORD3R	Moving rectangular coordinate system definition, form 1
CORD4R	Moving rectangular coordinate system definition, form 2
CORDROT	Corotational frame definition

Mesh Generation

BIAS	Non-uniform or biased mesh definition
MESH	Mesh generator

Lagrangian Elements

Solid Elements

CHEXA	Connection definition for brick element with eight grid points
CPENTA	Connection definition for wedge element with six grid points
CTETRA	Connection definition for tetrahedron element with four grid points
PSOLID	Property definition for CHEXA, CPENTA, CPENTA

Surface Elements

CQUAD4	Connection definition for a quadrilateral shell element with four grid points
CTRIA3	Connection definition for a triangular shell or membrane element with three grid points
PSHELL	Property definition for CQUAD4 and CTRIA3
PSHELL1	Extended property definition for CQUAD4and CTRIA3
PCOMP	Layered composite element property
PCOMPA	Additional data for layered composite element property

1-D Elements

CBAR	Connection definition for a line element with two grid points
CBEAM	Connection definition for a line element with two grid points
CROD	Connection definition for a line element with two grid points
CDAMP1	Connection definition for a scalar damper element with two grid points
CDAMP2	Connection definition for a linear damper element with two grid points
CELAS1	Connection definition for a scalar spring element with two grid points
CELAS2	Connection and property definition for a scalar spring element with two grid points
CSPR	Connection definition for spring element with two grid points
CVISC	Connection definition for a viscous damper element with two grid points
PBAR	Property definition for a CBAR element
PBCOMP	Alternate form of PBEAM
PBEAM	Property definition for CBAR and CBEAM
PBEAM1	Extended property definition for CBARand CBEAM
PBEAML	Complex property definition forCBAR and CBEAM by cross-sectional dimensions
PBELT	Property definition for a seat belt element defined by CROD
PDAMP	Property definition forCDAMP1 and CDAMP2
PELAS	Property definition for `CELASn`

PELAS1	Property definition on nonlinear elastic springs for CELASn
PROD	Property definition forCROD
PSPR	Property definition for CSPR
PSPR1	Property definition for nonlinear CSPR
PVISC	Property definition for CVISC
PVISC1	Property definition for nonlinear CVISC
PWELD	Property definition for spot welds with failure
PWELD1	Property definition for skin-stringer delamination (shell beam)
PWELD2	Property definition for sandwich structure delamination (solid shell)

Eulerian Elements

Solid Elements

CHEXA	Connection definition for a brick element with eight grid points
CPENTA	Connection definition for a wedge element with six grid points
CTETRA	Connection definition for a tetrahedral element with four grid points
PEULER	Property definition for CHEXA, CPENTA CTETRA
PEULER1	Property definition for CHEXA, CPENTA, CTETRA using geometrical regions

Constitutive Models

DMAT	General constitutive model
DMATEL	Isotropic elastic material properties
DMATEP	Elastic or elastoplastic material properties
DMATOR	Orthotropic material properties
DYMAT14	Soil and crushable foam material properties
DYMAT24	Piecewise linear plasticity material properties, with strain rate dependent plasticity
DYMAT25	Kinematic hardening Cap material properties
DYMAT26	Orthotropic crushable material properties used to model composites
FABRIC	Bi-directional woven fabric material properties
FOAM1	Crushable foam material properties
FOAM2	Crushable foam material properties
MAT1	Linear-isotropic material properties
MAT2	Anisotropic material properties

MAT8	Orthotropic elastic material properties used to model composites
MAT8A	Failure properties for orthotropic material properties
RUBBER1	Mooney-Rivlin model for rubber-like materials
RUBBER2	Ogden model for rubber-like materials
SHEETMAT	Anisotropic plasticity material used in sheet metal forming simulations

Yield Models

YLDEX	User-defined yield properties
YLDHY	Hydrodynamic yield properties
YLDVM	von Mises yield properties
YLDJC	Johnson-Cook yield properties
YLDMC	Mohr-Coulomb yield properties
YLDTM	Tanimura-Mimura yield properties
YLDZA	Zerilli-Armstrong yield properties
YLDRPL	Rate power law yield properties
YLDMSS	Snow material yield properties
YLDPOL	Polynomial yield properties
YLDSG	Steinberg-Guinan yield properties

Shear Models

SHREL	Elastic shear properties
SHRLVE	Isotropic linear viscoelastic shear properties
SHRPOL	Polynomial shear properties
SHREX	User-defined shear properties

Equations of State

EOSDEF	Deflagration equation of state
EOSMG	Mie-Gruneisen equation of state
EOSNA	Noble-Abel equation of state
EOSPOL	Polynomial equation of state
EOSJWL	JWL explosive equation of state
EOSGAM	Gamma law equation of state
EOSTAIT	Equation of state based on Tait model

EOSEX	Equation of state defined by user subroutines
EOSIG	Ignition and growth equation of state

Detonation Models

DETSPH	Spherical detonation wave

Failure Models

FAILEST	Maximum equivalent stress and minimum time-step failure model
FAILEX	Failure model defined by user subroutines
FAILEX1	Extended failure model defined by user subroutines
FAILEX2	User defined failure model using damage
FAILMES	Maximum equivalent stress failure model
FAILMPS	Maximum plastic strain failure model
FAILPRS	Maximum pressure failure model
FAILSDT	Maximum plastic strain and minimum time-step failure model
FAILJC	Johnson-Cook failure model

Spallation Models

PMINC	Constant spallation pressure properties

Rigid Bodies

MATRIG	Rigid-body properties
RBE2	Rigid-body element
RBHINGE	Rigid body hinge
RELEX	ATB ellipsoid to be used with Dytran
RELLIPS	Analytical rigid ellipsoid
RIGID	Rigid-body properties
SURFACE	Geometry of a rigid body

ATB Interface

ATBACC	Acceleration field applied to ATB segments
ATBJNT	Interface to ATB joints
ATBSEG	Interface to ATB segments

Lagrangian Constraints

Single-Point Constraints

GRDSET	Includes the default for single-point constraints on the GRID entry
GRID	Includes the single-point constraint definition (permanentSPCs)
SPC	Single-point constraint to put velocity components to zero
SPC1	Single-point constraint to put velocity components to zero
SPC2	Rotational velocity constraint
SPC3	Single-point constraint to put velocity components to zero in a local coordinate system

Contact Surfaces

CONTACT	Defines contact between Lagrangian objects
CONTFORC	Contact force definition using force-deflection curves
CONTINI	Defines initialization of contact state between two subsurfaces
CONTREL	Defines rigid-ellipsoid contact with Lagrangian grid points or rigid bodies
SURFACE	Defines a multifaceted surface
SUBSURF	Defines a multifaceted subsurface
CSEG	Defines segments of a surface
CFACE	Defines segments of a surface
CFACE1	Defines segments of a surface

Connections

JOIN	Defines a rigid connection between grid points of 1-D, shell, and solids
BJOIN	Defines a breakable rigid connection between grid points of 1-D and shell elements
KJOIN	Defines the kinematic join of shell and solid grid points
RCONN	Defines a rigid connection between two surfaces
RCONREL	Defines a connection with rigid ellipsoids for grids and surfaces
RJCYL	Cylindrical-joint constraint between rigid bodies
RJPLA	Planar-joint constraint between rigid bodies
RJREV	Revolute-joint constraint between rigid bodies
RJSPH	Spherical-joint constraint between rigid bodies
RJTRA	Translational-joint constraint between rigid bodies
RJUNII	Universal-joint constraint between rigid bodies

Rigid Walls

WALL	Defines a rigid wall for grid points

Rigid Body Constraints

RBC3	Three-point constraint on a rigid body
FORCE	Concentrated load or enforced translational velocity
MOMENT	Concentrated moment or enforced rotational velocity

Lagrangian Loading

Transient Loading

TLOAD1	Defines the transient load
TLOAD2	Defines the transient time-varying load
DAREA	Defines the position and scale factor of a concentrated load
FFCONTR	Closed volume intended for fluid filled containers
FORCE	Defines the position and scale factor of a concentrated force
FORCE1	Defines a follower force, form 1
FORCE2	Defines a follower force, form 2
MOMENT	Defines the position and scale factor of a concentrated moment
MOMENT1	Defines a follower moment, form 1
MOMENT2	Defines a follower moment, form 2
PLOAD	Defines the position and scale factor of a pressure load
PLOAD4	Defines the position and scale factor of a pressure load
RFORCE	Defines the centrifugal load
GRAV	Defines the gravitational load

Enforced Motion

BODYFOR	Defines a acceleration on grid points.
TLOAD1	Defines the transient enforced motion
TLOAD2	Defines the transient time-varying enforced motion
DAREA	Defines the direction and scale factor of motion
FORCE	Defines the direction and scale factor of motion

FORCE3	Defines the direction and scale factor of motion in local coordinate systems
FORCEEX	Enforced translational velocity defined by user subroutines
MOMENT	Defines the direction and scale factor of motion

Initial Conditions

TIC	Defines transient initial velocities of grid points
TIC1	Defines transient initial velocities of grid points
TIC2	Defines an initial rotational velocity field for grid points
TIC3	General Form of Transient Initial Velocities of Grid Points
TICEL	Defines the transient initial conditions of element variables
TICGP	Defines the transient initial conditions of grid point variables
TICEEX	Transient initial conditions of element variables defined by user subroutines
TICGEX	Transient initial conditions of grid-point variables by a user-written subroutine

Eulerian Loading and Constraints

Single-Point Constraints

ALEGRID	Defines the motion of Eulerian grid points
ALEGRID1	Definition of ALE motion for Eulerian grid points.
SPC	Single-point constraint to put velocities of ALEGRID points to zero
SPC1	Single-point constraint to put velocities of ALEGRID points to zero
SPC2	Rotational velocity constraint for ALEGRID points
SPC3	Single-point constraint to put velocity components to zero in a local coordinate system for ALEGRID points

Enforced Motion

BODYFOR	Defines a body force loading per unit mass.
BODYFR1	Defines a body force loading on specific region.

Flow Boundary

TLOAD1	Defines the transient load
FLOW	Defines the flow boundary
FLOWC	Defines cyclic flow boundary condition
FLOWCDR	Defines cyclic flow boundary condition for all faces in a specific direction

FLOWCSQ	Defines cyclic flow boundary condition using a square definition
FLOWDIR	Defines flow boundary condition for all faces in a specific direction
FLOWEX	Flow boundary defined by user subroutines
FLOWDEF	Defines the free Eulerian faces to be a flow boundary by default
FLOWSQ	Defines flow boundary condition using a square definition
FLOWT	Defines time-dependent flow boundary
FLOWTSQ	Defines time dependent flow boundary using a square definition
FLOWXDR	Defines user defined flow boundary in specific direction
FLOWXSQ	Defines user defined flow boundary condition using a square definition
PERMCPL	Defines permeability between coupling surfaces
PERMEAB	Defines air bag permeability using environmental variables
PERMGBG	Defines air bag permeability
POREX	Defines a porosity model through a user-written subroutine
PORFCPL	Defines flow between two coupling surfaces through a hole
PORFGBG	Defines flow between two air bags through a hole
PORFLCPL	Defines flow between two coupling surfaces through a large hole
PORFLGBG	Defines flow between two air bags through a large hole
PORFLOW	Defines a porous flow boundary
PORFLOWT	Defines time-dependent porous flow boundary
PORHOLE	Defines a hole in a couple and/or GBAG(sub)surface
PORHYDST	Defines hydrostatic pressure profile on porous surfaces
PORLHOLE	Defines a large hole in a couple and/or GBAG (sub)surface
CSEG	Defines the face to which the flow boundary is applied
CFACE	Defines the face to which the flow boundary is applied
CFACE1	Defines the face to which the flow boundary is applied

Wall

WALLDIR	Wall boundary condition for all Eulerian boundary faces in a specific direction.
WALLET	Defines a wall for Eulerian material flow.

Gravity

GRAV	Defines the gravitational load

Initial Conditions

TIC	Defines the initial rotational grid-point velocities for ALEGRIDpoints
TICGP	Defines the transient initial grid-point variables
TICEL	Defines the transient initial condition for element variables
TICEUL	Defines the transient initial conditions for element variables using geometric regions
TICVAL	Defines the transient initial conditions
INITGAS	Gas bag or coupling surface initial gas fraction definition
MATINI	Eulerian initialization surface
CYLINDER, SPHERE, BOX, BOX1	Defines the geometrical shapes

Hydrostatic Pressure

HYDSTAT	Hydrostatic Preset of Density in Euler Elements.

Euler/Lagrange Coupling

COUHTR	Heat transfer model to be used with COUPLE entry
COUINFL	Inflator model to be used with COUPLE entry
COUP1FL	Defines the surrounding variables when a segment of a coupling surface fails
COUP1INT	Defines the interaction between two coupling surfaces
COUPLE	Defines the general coupling between the Eulerian and Lagrangian meshes
COUPLE1	Defines the general coupling between the Roe solver for single hydro materials and Lagrangian structures
COUOPT	Defines the coupling options
COUPOR	Defines the coupling surface or subsurface porosity
ALE	Defines the arbitrary Lagrange-Euler (ALE) coupling
GBAG	Gas bag pressure definition
GBAGC	Defines flow between two gas bags
GBAGCOU	General coupling to gas bag switch to save CPU time
GBAGHTR	Heat transfer model to be used with GBAG entry
GBAGINFL	Inflator model to be used with GBAG entry
GBAGPOR	Gas bag porosity
HTRCONV	Air Bag Convection
HTRRAD	Air Bag Radiation

INFLATR	Air bag inflator model
INFLATR1	Extended air bag inflator model
INFLCG	Airbag cold gas inflator model
INFLFRAC	Hybrid inflator gas fraction definition
INFLGAS	Inflator gas definition
INFLHYB	Hybrid inflator model
INFLHYB1	Extended hybrid inflator model
INFLTANK	Air bag tank test inflator model
SURFACE	Defines the coupling surface
SUBSURF	Defines the subsurface

Miscellaneous

Comments

$	For inserting comments in Bulk Data Section

Parameters

PARAM	Specifies values for the parameters used in the solution

Tabular Input

TABLED1	Tabular functions for loads, properties, materials, etc.
TABLEEX	Analytical function for loads, properties, materials, etc. defined by user subroutines
TABFILE	Text file defined function

Sets

SET1	Sets of numbers for use by other entries
SETC	Sets of names for use by other entries

Solution Control

ACTIVE	Activates or deactivates elements and interaction
VISCDMP	Defines dynamic relaxation factors for damping

Output

SECTION	Cross section

Prestress Analysis

NASINIT	Defines the prestress analysis logistics

Input File Control

INCLUDE	Switches data input to another file

Bulk Data Control

BEGIN BULK	Marks the end of the Case Control and the beginning of Bulk Data
ENDDATA	Marks the end of the input data

Marker

CMARKB2	Two-noded marker connectivity definition
CMARKN1	One-noded marker connectivity definition
PMARKER	Property definition of a marker element

Default Setting

SETTING	Application-sensitive defaults

Hourglass Control

HGSUPPR	Hourglass Suppression Method

Ignore Definition

IGNORE	Ignore a Set of Euler Elements

Bulk Data Descriptions

The format of each Bulk Data entry and the contents of each field on the entry is described here. The **Type** column indicates the type of data in the field. This can be `I` (Integer), `R` (Real), or `C` (Character). In addition, there may be limits on the value that can be entered in the field. For example, `I > 0` indicates that you must supply an integer with a value greater than zero. The value limitation $0 < R \leq 1$ indicates that you must supply a real number greater than zero and less than or equal to one.

The **Default** column indicates the value that is used if the field is left blank. If the word “Required” appears, there is no default and you must supply a value.

$ Comment

Anything that appears after a $ on a line is treated as a comment and is ignored. If a $ appears as the first character, the entire line is a comment.

Format and Example

1	2	3	4	5	6	7	8	9	10
$ followed by any characters on the rest of the line									
$ THE WHOLE LINE IS A COMMENT.CONTACT									
GRID	1		0.0	10.0	130.0	$ THE REST OF THE LINE IS A COMMENT.			

Remark

If a comment is placed in fields which would otherwise contain data, the data in those fields is given the fields' default values.

ACTIVE

Activate Elements and Interaction

Allows you to activate parts of the program for a part of the problem time only.

Format and Example

1	2	3	4	5	6	7	8	9	10
ACTIVE	ID	TYPE	TYPEV	INITV					+
ACTIVE	3	INTERACT	COUPLE						+
+	TIME	TIMEV							
+	TABLE	1							

Field	Contents			Type	Default
ID	Unique active number			I > 0	Required
TYPE	Type of activity switch.			C	Required
		ELEMENT	Switches are for the element type as defined under TYPEV.		
		INTERACT	Switches are for an algorithm defining the interactions between different parts of the model. The type of algorithm is defined under TYPEV.		
		RIGID	Switches are for rigid entities as defined under TYPEV.		
TYPEV	Depends on the value of TYPE:			C	Required
		TYPE	TYPEV		
		ELEMENT	SHTRIA SHTRIA SHQUAD MEMTRIA DUMTRIA DUMQUAD LAGSOLID EULHYDRO EULSTRENGTH MULTIEULHYDRO ELEM1D		
		INTERACT CONTACT	COUPLE GBAG		
		RIGID SURFACE			
INITV	NO:	Element Initalization at cycle 0 (default).			NO

Field	Contents				Type	Default
	YES:	Element Initialization when the elements become active				
TIME	Type specification for switches.				C	Required
		TABLE	Part is switched on and off, depending on the y-value of the table with ID as specified in TIMEV. The x-value of the table represents the time; the y-value means:			
			ON	y > 0		
			OFF	y ≤ 0		
TIMEV	Number of a TABLED1 or TABLEEX				I > 0	Required

Remarks

1. The default is all parts of the program are active at all times.
2. For CONTACT, an activity switch can also be set on the entry itself. These settings overrule settings on the ACTIVE entry.
3. The active option for multimaterial with shear strength is activated by using TYPEV = MULTIEULHYDRO.
4. For COUPLE and C in combination with PARAM, LIMITER, ROE, an activity switch can also be set on the COUPLE1 entry. These settings overrule the settings on the ACTIVE entry.
5. INITV is only used when TYPE is EULHYDRO, EULSTRENGTH, or MULTIEULHYDRO. INITV is ignored for all other types. For simulations involving a prestress phase, the Euler elements can be initially deactivated. When the Euler elements are activated, the Eulerian masses still originate from the Euler initiation at cycle 0. If during prestressing, the structure did not move much, then these Eulerian masses of cycle 0 can give a stable run. But if there has been substantial movement of the structure, then Eulerian masses can be erroneously compressed. This shows up by large pressures and velocities in the Euler elements and a time step too small. To avoid this instability, the Euler initialization can be postponed until the Euler elements become active.

ALE

Arbitrary Lagrange Eulerian (ALE) Interface

Defines the surfaces of an ALE interface.

Format and Example

1	2	3	4	5	6	7	8	9	10
ALE	AID	SIDLG	SIDEU						
ALE	32	3	5						

Field	Contents	Type	Default
AID	Unique ALE interface number.	I > 0	Required
SIDLG	Number of a SURFACE entry that defines the Lagrangian part of the ALE interface.	I > 0	Required
SIDEU	Number of a SURFACE entry that defines the Eulerian part of the ALE interface.	I > 0	Required

Remarks

1. SIDLG and SIDEU must reference the SID of a SURFACE entry.
2. The Eulerian and Lagrangian SURFACEs must have a one-to-one correspondence. This means that the Eulerian and Lagrangian grid points in the SURFACEs must coincide in physical but not in logical space (same position, different number).
3. The tolerance used in finding coinciding SURFACE nodes is defined by the ALETOL parameter.
4. ALE is not applicable in combination with the single material Euler solver with a full-stress tensor. (EULSTRENGTH), please use the multimaterial solver instead of (MULTIEULSTRENGTH).

ALEGRID

Eulerian Grid Point Motion Definition

Definition of ALE motion for Eulerian grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
ALEGRID	AID	MINCUT	MAXCUT	TYPE	WEIGHT	NAME			+
ALEGRID	28	0.	1.	STANDARD	COMPUTED				+
+	G1	G2	THRU	G3	BY	G4	-etc.-		
+	1	2	THRU	15	BY	3			

Field	Contents	Type	Default
AID	ALEGRID number	I > 0	Required
MINCUT	See Remark 1.	R	0
MAXCUT	See Remark 1.	R	1.E20
TYPE	Indicates the type of motion. (See Remark 2.)	C	SPECIAL
	STANDARD		
	FREE		
	FIXED		
	FLOW		
	SPECIAL		
	USER		
WEIGHT	Method of calculating weight factors. (See Remark 6.)	C	COMPUTED
	EQUAL		
	COMPUTED		
NAME	Name of the motion prescription passed to the user subroutine. See Remark 2.	C	None
G1,G2...	Grid points to which the motion applies. THRU indicates the range, while BY allows an increment to be used within this range.	I > 0	Required

Remarks

1. The MINCUT and MAXCUT parameters define the minimum and maximum allowable grid-point velocity of ALE grid points. Usually the defaults are sufficient.

$$\vec{u}_g = max\left(\text{MINCUT}\frac{\Delta q}{\Delta t}, \left|\vec{u}_g\right|\right) \cdot sign(\vec{u}_g)$$

$$\vec{u}_g = min\left(\text{MINCUT}\frac{\Delta q}{\Delta t}, \left|\vec{u}_g\right|\right) \cdot sign(\vec{u}_g)$$

   where Δq is the element characteristic dimension and Δt is the time step.

2. The TYPE definition causes the grid point motion algorithm to define grid point velocities as follows:

STANDARD:	Each grid point moves to the center of its neighbors.
FREE:	The grid points that are defined as FREE move as on a free surface. The grid point velocity becomes $\vec{u}_g = \vec{u}_{g_{tentative}} + [(\vec{u}_{fs} - \vec{u}_{g_{tentative}}) \cdot \vec{n}] \cdot \vec{n}$ where $\vec{n}$ is the normal to the free surface. $\vec{u}_{fs}$ is the free surface velocity defined as: $\vec{u}_{fs} = \frac{\sum_{i=1} \vec{v}_i}{N}$ with $\vec{v}_i$ the material velocity of the elements connected to the grid point. $\vec{u}_{g_{tentative}}$ is the tentative grid point velocity.
FIXED:	Grid points that are defined as FIXED move as on a fixed wall. The grid point velocity becomes $= u_{g_{tentative}} - (u_{g_{tentative}} \cdot \vec{n} \cdot i$ where $\vec{n}$ is the normal to the wall.
FLOW:	Grid points move as on a flow boundary. The grid point velocity becomes $\vec{u}_g = \vec{u}_{g_{int}} + [(\vec{u}_{g_{tentative}} - \vec{u}_{g_{int}}) \cdot \vec{t}] \cdot \vec{t}$ where $\vec{g}_{int}$ is the grid point velocity of the closest internal grid point. The vector tangent to the flow boundary is given by $\vec{t}$.

SPECIAL	Dytran searches the grid points defined on the ALEGRID entry. It detects which surface boundary condition the grid points are part of. The grid point motion is corrected correspondingly.
USER:	The grid point motion is defined via the user-written subroutine. The name that is defined in the NAME field can be used to distinguish different motion prescriptions in the user subroutine.

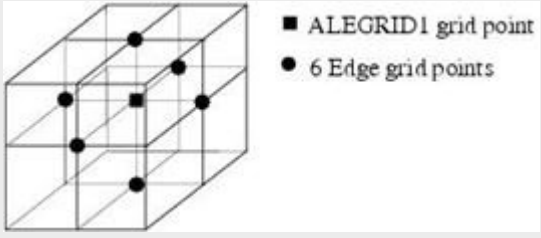
3. More than one ALEGRID entry can occur in input, with each one having a different type definition. All ALEGRID entries that have the same AID are merged into one definition. This requires a consistent definition with respect to the options of all of the ALEGRID entries that have the same AID.
4. The number of relaxation iterations for the grid point displacement is one by default but can be changed using PARAM,ALEITR.
5. There can be as many continuation lines as necessary.
6. The weight factors determine the grid point motion. If the option is set to COMPUTED (default), Dytran computes the weight factors based on geometrical considerations. If the option is set to EQUAL, all weight factors are set to a constant. The latter is automatically done when a local distortion of the Eulerian mesh is encountered that does not allow for the computation of the weight factors.
7. If the TYPE field is set to USER, all other fields are ignored except the NAME field which is mandatory.
8. For a description of user-written subroutines, see Chapter 7: User Defined Services in this manual.

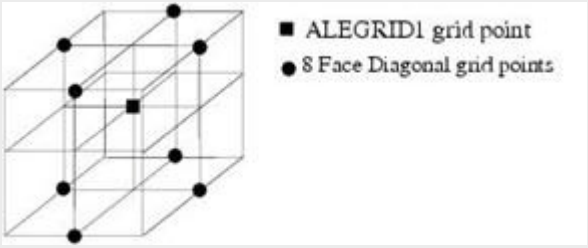
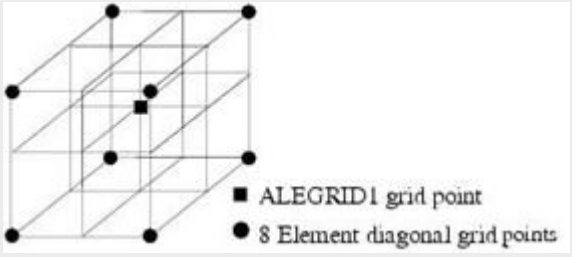
ALEGRID1 Eulerian Grid Point Motion Definition

Definition of ALE motion for Eulerian grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
ALEGRID1	AID	SID	EDGE	FDIAG	EDIAG				+
ALEGRID1	28	11	YES	NO	NO				+

Field	Contents	Type	Default
AID	ALEGRID1 number.	I > 0	Required
SID	Number of SET1 grid point entries (see Remark 3.)	I > 0	Required
EDGE	YES/NO Specifies if the neighboring grid points along the edges need to be taken into account for the mesh motion. 	C	YES

Field	Contents	Type	Default
FDIAG	YES/NO Specifies if the neighboring grid points along the face diagonals need to be taken into account for the mesh motion.  Adding these nodes increases the required amount of memory to store the data, but might improve the motion of the mesh.	C	NO
EDIAG	YES/NO Specifies if the neighboring grid points along the element diagonals need to be taken into account for the mesh motion.  	C	NO

Remarks

1. The ALEGRID1 mesh motion algorithm has the following features:

   No limit on the number of neighboring grid points exists. This is in contrast to the ALEGRID algorithm where a limit of eight neighboring nodes exists.

   The velocity of the grid points is based on a changing weight factor, which is calculated accordingly

$$= \frac{\sum(\Delta l_{g_i} \times u_{\xi}}{\sum \Delta l_{\sigma}}$$

u_g	=	the velocity component of the Eulerian grid point
u_{g_i}	=	the velocity component of the neighbor grid points i
Δl_{g_i}	=	the distance between the Eulerian grid point and the neighbor grid points i

2. When the Lagrangian mesh is moving very fast it can happen that the Eulerian mesh is not properly following the structure, and Eulerian elements get distorted.

   The mesh motion can be improved by increasing the value of `PARAM,ALEITR`. Multiple iterations per time step will be performed to determine the grid point velocities.

3. Multiple `SET1` entries with the same `SID` are automatically concatenated.
4. A combination of ALEGRID and `ALEGRID1` entries in one analysis model is not allowed.

ATBACC

Acceleration Field Applied to ATB Segments

Defines an acceleration field that will be applied to ATB segments.

Format and Example

1	2	3	4	5	6	7	8	9	10
ATBACC	LID		SCALE	NX	NY	NZ			+
ATBACC	32		386.088	1.0	0.0	0.0			+
+	NAME1	NAME2	NAME3	NAME4	NAME5	NAME6	NAME7	-etc-	
+	LT	MT	UT	N	H	RUL	RLL		

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
SCALE	ATBACC scale factor	R ≥ 0.0	1.0
NX, NY, NZ	Components of gravity vector. At least one component must be nonzero.	R ≥ 0.0	0.0
NAMEi	Name of an ATB segment as given in the first field of a B.2 entry in the ATB input file.	C	Required

Remarks

1. The acceleration $a(t)$ is defined as:

   $$a(t) = T(t)*\text{SCALE}*\text{N}$$

   where SCALE is the acceleration scale factor; N is the vector defined by NX, NY, and NZ; $T(t)$ is the value interpolated at time t from the table referenced by the TLOADn entry.
2. LID must be referenced by a TLOADn entry.
3. The type field on the TLOADn entry must be set to zero.
4. More than one ATBACC acceleration field can be defined per problem.
5. This acceleration field is intended to apply a crash pulse to ATB segments that define a crash dummy. The acceleration is multiplied by the mass of the segment and the resulting force is added as an external force.
6. To compare the accelerations of the ATB segments to experiments, the crash pulse needs to be subtracted from the total acceleration. The acceleration of the segments as defined on the H1 entries in the ATB input file are automatically corrected.

ATBJNT

Interface to ATB Joints

This entry can only be used together with the ATBSEG entries that this joint connects. The ATBSEG entries overwrite the position and orientation of the ATB segments as specified in the ATB input file. The ATBJNT entry can be used to create a Bulk Data file containing elements and grid points to visualize the ATB segment together with its joints. This visualization of the joints makes it possible to position the ATB model in any available preprocessor. See also PARAM,ATBSEGCREATE.

Format and Example

1	2	3	4	5	6	7	8	9	10
ATBJNT	ID	NAME							+
ATBJNT	1	HN							+
+	G0	G1	G2	G3	EID1	EID2	EID3		+
+	1010	1011	1012	1013	1004	1005	1006		+
+	G4	G5	G6	G7	EID4	EID5	EID6		
+	2010	2011	2012	2013	2004	2005	2006		

Field	Contents		Type	Default
ID	Unique ATBJNT number		I > 0	Required
NAME	Name of an ATB joint as given in the first field of a B.3 entry in the ATB input file		C	Required
G0-G3 G4-G7	An ATB joint connects two segments. A local joint coordinate system is attached to each of these two segments. The position and orientation of these two coordinate systems relative to the segment coordinate systems is given on entry B.3 in the ATB input file. For each joint (J = 1,NJNT) a B.3 entry is defined in the ATB input file. The joint J connects the segment JNT(J) as given on the B.3 entry and the segment J + 1. Dytran finds the two segments that are connected by the joint with name = NAME. The grid points G0-G3 and G4-G7 define the joint coordinate systems for the segments JNT(J) and J + 1, respectively:		I > 0	Required
	G0	located at the origin of the joint coordinate system for the ATB segment JNT (J)		
	G1	located on the local x-axis.		
	G2	located on the local y-axis.		
	G3	located on the local z-axis.		

Field	Contents	Type	Default
	G4 located at the origin of the joint coordinate system for the ATB segment J + 1.		
	G5 located on the local x-axis.		
	G6 located on the local y-axis.		
	G7 located on the local z-axis.		
EID1-EID3 EID4-EID6	If EID1 through EID6, and PARAM,ATBSEGCREATE have been specified, Dytran generates a Bulk Data file at time = 0. The grid points G0-G3 and G4-G7, at their initial position as specified in the ATB input file, are written to the file. The files also contain the following CBAR entries:	I > 0	Blank
	For segment JNT(J):		
	CBAR, EID1, PID-JNT(J), G0, G1, G2		
	CBAR, EID2, PID-JNT(J), G0, G2, G3		
	CBAR, EID3, PID-JNT(J), G0, G3, G1		
	For segment J+1		
	CBAR, EID4, PID-(J + 1), G4, G5, G6		
	CBAR, EID5, PID-(J + 1), G4, G6, G7		
	CBAR, EID6, PID-(J + 1), G4, G7, G4		

Remark

All elements related to an ATB segment refer to the same material number. This material number is defined on the ATBSEG entry. If the material is defined to be rigid by means of a MATRIG entry, all elements can be easily connected to the contact ellipsoid of the ATB segment by means of an RCONREL entry referencing the MATRIG entry. In this way, all elements related to an ATB segment move together with the ATB segment during the analyses and can be postprocessed.

ATBSEG

Interface to ATB Segments

Defines the position and orientation of the ATB segments. The position and orientation as specified on the G.2 and G.3 entries in the ATB input file will be overruled by the definitions given here.

This entry can be used to create a Bulk Data file containing elements and grid points to visualize the ATB segment, together with the contact ellipsoid and the joints it is connected by. See also ATBJNT and PARAM,ATBSEGCREATE.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ATBSEG	ID	NAME	COVER	NUMELM	GSTART	ESTART	MID	PIDCOV	+
ATBSEG	1	HEAD	YES	100	1000	1000	1000	1000	+
+	G0	G1	G2	G3	EID1	EID2	EID3	PIDCG	
+	1010	1001	1002	1003	1001	1002	1003	1001	

Field	Contents	Type	Default
ID	Unique ATBSEG number	I > 0	Required
NAME	Name of an ATB segment as given in the first field of a B.2 entry in the ATB input file	C	Required
COVER	YES If PARAM,ATBSEGCREATE has been specified, Dytran generates a Bulk Data file containing grid points and elements located on the surface of the segment contact ellipsoid. The shape and position of the segment contact ellipsoid is defined on the B.2 entry in the ATB input file. See Remark 2. NO The covering is not performed.	C	NO
NUMELM	Maximum number of elements used for covering the ellipsoid.	I > 0	128
GSTART	Grid-point numbering for covering the ellipsoid starts at GSTART.	I > 0	1
ESTART	Element numbering for covering the ellipsoid starts at ESTART.	I > 0	1
MID	All elements created by Dytran to visualize the ATB segment will have a rigid material MATRIG) with MID as the material number. MID is used by both the elements covering the segment contact ellipsoid as well as by the CBAR elements used to visualize the segment coordinate system and joint coordinate systems (See ATBJNT).	I > 0	1
PIDCOV	All elements created by Dytran to cover the ATB segment contact ellipsoid will have PIDCOV as the property number.	I > 0	1

Field	Contents	Type	Default
G0-G3	The grid points span the local coordinate system of the ATB segment: G0: located at the origin of the ATB segment. G1: l located on the local x-axis. G2: located on the local y-axis. G3: located on the local z-axis. The above is used by Dytran to overwrite the initial position and orientation of the segments as specified in the ATB input file. See below (EID1-EID3) on how to generate the above grid points for an existing ATB input file.	I > 0	Required
EID1-EID3	If EID1, EID2, EID3, and PARAM,ATBSEGCREATE have been specified, Dytran generates a Bulk Data file containing the grid points G0-G3 at the initial position as specified in the ATB input file. The file also contains the three following CBAR entries: CBAR, EID1, PIDCG, G0, G1, G2 CBAR, EID2, PIDCG, G0, G2, G3 CBAR, EID3, PIDCG, G0, G3, G1	I > 0	Blank
PIDCG	Property number used by Dytran in generating the CBAR entries EID1 through EID3.	I > 0	1

Remarks

1. All elements related to an ATB segment reference the same material number. This material number is defined on the ATBSEG entry. If the material is defined as rigid by means of a MATRIGentry, all elements can be easily connected to the contact ellipsoid of the ATB segment by means of an RCONREL entry referencing the MATRIGentry. In this way, all elements related to an ATB segment move together with the ATB segment during the analysis and can be postprocessed. The elements can also be used in a CONTACT ALE, and/or COUPLING surface to define interaction between the ATB segment and other parts of the finite element model. The forces and moments acting on the elements are transferred to the ATB segment to which they are connected.
2. The MATRIG entry written to the file has the inertia properties of the segment, as defined in the ATB input file.

BARLT89

Sheet-Metal Material

Defines the properties of an anisotropic plastic material under plane stress condition. This material model uses the yield criterion developed by Barlat and Lian [1989], Barlat-89, which allows the use of Lankford parameters to define material anisotropy. This material model is only applicable for Lagrangian shell elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BARLT89	MID	RHO	E			G			+
BARLT89	1	2.7E-6	72E6						+
+	NU			ELASTIC		XMAT	YMAT	ZMAT	+
+	0.33			ISO		1.0	0.0	0.0	+
+	INPYLD	YLDM	YLDA Or SIGBAR	YLDC Or SIG90	YLDH Or TS1	YLDP Or TS2			+
+	3	6							+
+	HRDA	HRDB	HRDC	HRDN	HRDK	HRDM	HRDTBL		+
+	0.0	570E3	0.017	0.359	0.014	0.389			+
+	TYPEYLD	R00	R45	R90		CPTOL			+
+	PLANANI	0.73	0.5	0.69		1.0e-3			+
+	TYPEHRD								+
+	ISO								+
+	C1	C2	C3	C4	C5				+
+	0.244	-0.195	0.857	3.439	-11.9				+
+		D2	D3	D4	D5				+
+		-0.417	-1.57	-4.849	-6.06				+

Field	Contents		Type	Default
MID	Unique material number		I>0	Required
RHO	Mass density		R>0.0	Required
E	Young's moduli		R>0.0	See Remark 2.
G	Shear modulus		R>0.0	See Remark 2.
NU	Poisson's ratios		R≥0.0	See Remark 2.
ELASTIC	Type of elasticity		C	ISO See Remark 2.
	ISO	ISOtropic material		
XAMT, YMAT, ZMAT	Vector indicating the rolling direction of the material		R	(0,0,0) See Remark 3.
INPYLD	Yield parameter input option.		I	See Remark 4.
	1	Input Baralat-89 yield criterion parameters directly.		
	2	Baralat-89 yield criterion parameters are calculated using uniaxial, biaxial, and shear tests data.		
	3	Baralat-89 yield criterion parameters are calculated using Lankford parameters.		
YLDM, YLDA, YLDC, YLDH, YLDP	Baralat-89 yield criterion parameters		R≥0.0	See Remark 4.
SIGBAR	Yield stress for uniaxial tension in the rolling direction		R≥0.0	See Remark 4.
SIG90	Yield stress for uniaxial tension in transverse to the rolling direction		R≥0.0	
TS1	Yield stress for biaxial test		R≥0.0	
TS2	Yield stress for pure shear test		R≥0.0	
HRDA	Power-law stress constant		R≥0.0	Required See Remark 4.
HRDB	Power-law hardening parameter		R≥0.0	0.0
HRDC	Power-law strain offset		R≥0.0	0.0
HRDN	Power-law, strain-hardening exponent		R≥0.0	1.0

Field	Contents		Type	Default
HRDK	Power-law, strain-rate sensitivity constant		R≥0.0	0.0 See Remark 5.
HRDM	Power-law, strain-rate exponent		R≥0.0	1.0
HRDTBL	Number of a TABLED1 entry giving the variation of yield stress (y-value) with effective plastic strain (x-value). If nonzero and positive, the curve will be used to define yield stress.		I>0	
TYPEYLD	Type of yielding criterion		C	ISO See Remark 6.
	ISO	ISOtropic yielding		
	NORMANI	NORMal ANIsotropic yielding		
	PLANANI	PLANar ANIsotropic yielding		
R00, R45, R90	Anisotropic yielding parameters (Lankford parameters) defined in 0, 45, and 90 degrees with respect to the rolling direction		R>0.0	See Remark 7.
CPTOL	Convergence tolerance for cutting plane algorithm		R≤1.0e-2	1.0e-6
TYPEHRD	Type of hardening rule		C	ISO
	ISO	ISOtropic hardening		
	NORMANI	NORMal ANIsotropic hardening		
C1-C5	Engineering coefficients in limit function for e2 > 0.		R	C1=1.0 See Remark 8.
D2-D5	Engineering coefficients in limit function for e2 < 0.		R	0.0 See Remark 8.

Remarks

1. BARLT89 materials may only be referenced by PSHELL and PSHELL1 entries.
2. BARLT89 material model supports only isotropic elasticity (ELASTIC=ISO), hence only E and G (or NU) must be defined.
3. Material anisotropy can be defined by providing the rolling direction as the reference direction.
   - XHAT, YHAT, ZHAT = (0.,0.,0.): Base axis is node 1 to node 2 in the element connectivity table + THETA on the CQUAD4 and CTRIA3 entries.

- XHAT, YHAT, ZHAT: ≠ (0.,0.,0.): The projection of the vector (XMAT, YMAT, ZMAT)on the surface of each element is used to determine the angle between the element and the material coordinate system. This angle can be overwritten using the THETA field on the CQUAD4 and CTRIA3 entries.

Both the constitutive law and output of variables are applied with respect to this material coordinate system (see Remark 9.).

For example:

Case	CTRIA3/CQUAD4 THETA	XMAT/YMAT/ZMAT	Rolling Direction
1	THETA	(X,Y,Z)	THETA in CTRIA3/CQUAD4
2	0.0	(X,Y,Z)	(X,Y,Z) in XMAT/YMAT/ZMAT
3	Blank	(X,Y,Z)	(X,Y,Z) in XMAT/YMAT/ZMAT
4	THETA	(0,0,0)	THETA in CTRIA3/CQUAD4
5	0.0	(0,0,0)	0.0 in CTRIA3/CQUAD4
6	Blank	(0,0,0)	0.0 in CTRIA3/CQUAD4

4. For a description of the BARLT89 model including the yield criterion, see *Dytran Theory Manual*, Chapter 3: Materials. INPYLD provides you with three options to define the yield surface. The hardening function can be introduced by providing the variation of yield stress with respect to the effective plastic strain. This relationship can be given using either a table or the power-law function and not both. The power-law stress constant, HRDA, is not necessarily the initial yield stress: the value of HRDA is allowed to be equal to zero if the value of the hardening parameter, HRDB, and the strain offset, HRDC, are unequal to zero.
5. Strain-rate dependency is not accounted for if HRDK and HRDM are left blank.
6. The field TYPEYLD provides you with an input check on the consistency of the anisotropic yielding parameters. Normal anisotropic material, NORMANI, is equivalent to transversely anisotropic or planar isotropic material which means that the through-the-thickness yielding properties may differ from the in-plane, isotropic, yielding properties. Planar anisotropic material, PLANANI, is characterized by three orthogonal axes of anisotropy (in rolling, transverse, and through-the-thickness direction), about which the yielding properties have twofold symmetry.
7. The necessary number of anisotropic-yielding parameters depends on the field TYPEYLD. For TYPEYLD = ISO, all fields for R00, R45, and R90 can be left blank as the default corresponds to R0 = R45 = R90 = 1.0. For TYPEYLD = NORMANI, only R00 must be defined while the other two fields can be left blank due to their equality (R0 = R45 = R90). The input of all three anisotropic parameters is needed for TYPEYLD = PLANANI.
8. C1 through C5 and D2 through D5 do not affect the material behavior but are used to fit the lower bound of experimental results for diffuse and localized necking represented by two polynomial lines:

$$FLD(e_2) = C_1 + C_2 e_2 + C_3 e_2^2 + C_4 e_2^3 + C_5 e_2^4 \text{ for } e_2 > 0.0$$

$$FLD(e_2) = C_1 + D_2 e_2 + D_3 e_2^2 + D_4 e_2^3 + D_5 e_2^4 \text{ for } e_2 < 0.0$$

9. The output of variables related to BARLT89 is defined with respect to the material coordinate system (see Remark 4.). There are a number of specific output variables useful for this material:

Element Variables	
Q1, Q2	Direction cosines/sines between the element coordinate system and the material coordinate system
Sublayer Variables	
TXX	Stress - XX component
TYY	Stress - YY component
TXY	Stress - XY component
TYZ	Stress - YZ component
TZX	Stress - ZX component
EFFST	Effective Stress
EFFPL	Effective Plastic Strain
YLDRAD	Radius of Yield Surface
EPSXX	Strain - XX component
EPSYY	Strain - YY component
EPSXY	Strain - XY component
EPSZZ	Strain - ZZ component
EPZZ	Plastic Strain – ZZ component
EPSMX	Strain - Major Principal Strain
EPSMN	Strain - Minor Principal Strain
FLP	Forming-Limit Parameter

BARLT00

Barlat YLD-2000 material

Defines the properties of an anisotropic plastic material under plane stress condition. This yield criterion was first introduced by Barlat, et al. [2003] (YLD2000-2D). It allows to define material anisotropy in terms of Lankford parameters and yield stress data. This material model is only compatible with Lagrangian shell elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
BARLT00	MID	RHO	E			G			+
BARLT00	1	2.7E-6	7.2E6						+
+	NU			ELASTIC		XMAT	YMAT	ZMAT	+
+	0.33			ISO		1.0	0.0	0.0	+
+	INPYLD	YLDM	CPTOL						+
+	1	8	1.0E-3						+
+	ALPHA1 or SIG0	ALPHA2 or SIG45	ALPHA3 or SIG90	ALPHA4 or SIG_b	ALPHA5 or R00	ALPHA6 or R45	ALPHA7 or R90	ALPHA8 or R_b	+
+	1.01	1.02	0.98	1.08	0.78	0.56	0.87	0.48	+
+	HRDA	HRDB	HRDC	HRDN	HRDK	HRDM	HRDTBL		+
+	0.0	570E3	0.017	0.359	0.014	0.389			+
+	TYPEHRD								+
+	ISO								+
+	C1	C2	C3	C4	C5				+
+	0.244	-0.195	0.857	3.439	-11.9				+
+		D2	D3	D4	D5				
+		-0.417	-1.57	-4.849	-6.06				

Field	Contents		Type	Default
MID	Unique material number		I>0	Required
RHO	Mass density		R>0.0	Required
E	Young's moduli		R>0.0	See Remark 2.
G	Shear modulus		R>0.0	See Remark 2.
NU	Poisson's ratios		R≥0.0	See Remark 2.
ELASTIC	Type of elasticity		C	ISO See Remark 2.
	ISO	ISOtropic material		
XAMT, YMAT, ZMAT	Vector indicating the rolling direction of the material		R	(0,0,0) See Remark 3.
INPYLD	Yield parameter input option.		I	See Remark 4.
	1	Input Baralat YLD2000 yield criterion alpha parameters directly		
	2	YLD2000-2D anisotropy parameters are calculated using directional and biaxial yield stress data (y0, y45, y90, yb) and corresponding Lankford parameters (r0, r45, r90, rb).		
YLDM	Exponent term in flow rule.		R≥0.0	See Remark 4.
CPTOL	Convergence tolerance for cutting plane algorithm		R≤1.0E-2	1.0E-6
SIG0	Normalized yield stress at 0 degree in the rolling direction.		R≥0.0	See Remark 4.
SIG45	Normalized yield stress at 45 degree in the rolling direction.		R≥0.0	See Remark 4.
SIG90	Normalized yield stress at 90 degree in the rolling direction.		R≥0.0	See Remark 4.
SIG_b	Normalized yield stress at biaxial test.		R≥0.0	See Remark 4.
R0, R45, R90	Anisotropic yielding parameters (Lankford parameters) defined in 0, 45, and 90 degrees with respect to the rolling direction		R>0.0	See Remark 4.
R_b	Anisotropic yielding parameters (Lankford parameters) defined in biaxial test		R>0.0	See Remarks 4. and 5.
ALPHA1~8	Barlat YLD2000 material parameters		R≥0.0	See Remark 4.
HRDA	Power-law stress constant		R≥0.0	See Remark 4.

Field	Contents		Type	Default
HRDB	Power-law hardening parameter		R≥0.0	0.0
HRDC	Power-law strain offset		R≥0.0	0.0
HRDN	Power-law, strain-hardening exponent		R≥0.0	1.0
HRDK	Power-law, strain-rate sensitivity constant		R≥0.0	0.0
HRDM	Power-law, strain-rate exponent		R≥0.0	1.0
HRDTBL	Number of a TABLED1 entry giving the variation of yield stress (y-value) with effective plastic strain (x-value). If nonzero and positive, the curve will be used to define yield stress.		I ≥0	See Remark 4.
TYPEHRD	Type of hardening rule.		C	ISO
	ISO	ISOtropic hardening		
	NORMANI	NORMal ANIsotropic hardening		
C1-C5	Engineering coefficients in limit function for e2 > 0.		R	C1=1.0, not used
D2-D5	Engineering coefficients in limit function for e2 < 0.		R	Not used

Remarks:

1. BARLT00 may only be referenced by PSHELL and PSHELL1 entries.
2. BARLT00 material model supports only isotropic elasticity (ELASTIC=ISO), hence only E and G (or NU) must be defined.
3. Material anisotropy can be defined by providing the rolling direction as the reference direction.
   - XHAT, YHAT, ZHAT = (0.,0.,0.): Base axis is node 1 to node 2 in the element connectivity table + THETA on the CQUAD4 and CTRIA3 entries.
   - XHAT, YHAT, ZHAT: ≠ (0.,0.,0.): The projection of the vector (XMAT, YMAT, ZMAT) on the surface of each element is used to determine the angle between the element and the material coordinate system. This angle can be overwritten using the THETA field on the CQUAD4 and CTRIA3 entries.

   Both the constitutive law and output of variables are applied with respect to this material coordinate system (see Remark 8.).

   For example:

Case	CTRIA3/CQUAD4 THETA	XMAT/YMAT/ZMAT	Rolling Direction
1	THETA	(X,Y,Z)	THETA in CTRIA3/CQUAD4
2	0.0	(X,Y,Z)	(X,Y,Z) in XMAT/YMAT/ZMAT
3	Blank	(X,Y,Z)	(X,Y,Z) in XMAT/YMAT/ZMAT

Case	CTRIA3/CQUAD4 THETA	XMAT/YMAT/ZMAT	Rolling Direction
4	THETA	(0,0,0)	THETA in CTRIA3/CQUAD4
5	0.0	(0,0,0)	0.0 in CTRIA3/CQUAD4
6	Blank	(0,0,0)	0.0 in CTRIA3/CQUAD4

4. For a description of the BARLT00 model including the yield criterion, see Materials (Ch. 3) in the *Dytran Theory Manual*. INPYLD provides you with two options to define the yield surface. The hardening function can be introduced by providing the variation of yield stress with respect to the effective plastic strain. This relationship can be given using either a table or the power-law function. The power-law stress constant, HRDA, is not necessarily the initial yield stress: the value of HRDA is allowed to be equal to zero if the value of the hardening parameter, HRDB, and the strain offset, HRDC, are unequal to zero.
5. R_b is optional. When it is set to empty, an additional constraint is applied, see Materials (Ch. 3) in the *Dytran Theory Manual*.
6. Strain-rate dependency is not accounted for if HRDK and HRDM are left blank.
7. C1 through C5 and D2 through D5 do not affect the material behavior but are used to fit the lower bound of experimental results for diffuse and localized necking represented by two polynomial lines:

$$FLD(e_2) = C_1 + C_2 e_2 + C_3 {e_2}^2 + C_4 {e_2}^3 + C_5 {e_2}^4 \qquad for \qquad e_2 \geq 0.0$$

$$FLD(e_2) = C_1 + D_2 e_2 + D_3 {e_2}^2 + D_4 {e_2}^3 + D_5 {e_2}^4 \qquad for \qquad e_2 < 0.0$$

8. The output of variables related to BARLT00 is defined with respect to the material coordinate system (see Remark 4.). There are several specific output variables useful for this material.

BEGIN BULK

The Beginning of the Bulk Data

Marks the end of the Case Control Section and the beginning of the Bulk Data Section in the input file.

Format and Example

```
BEGIN BULK
```

Remark

A `BEGIN BULK` entry must always be present.

BIAS

Bias Definition

Specifies a variation of the mesh-size in one direction for use in the MESH entry. The MESH entry can create a biased or non-uniform mesh. A uniform mesh consists of a number of planes separated by a fixed distance, but for a biased mesh the distance between subsequent planes can differ. The BIAS definition allows specifying the locations of planes in one direction. For a number of intervals the density of planes can be specified.

Format and Example

1	2	3	4	5	6	7	8	9	10
BIAS	ID								+CONT1
BIAS	100								+CONT1
+CONT1	X0	GROWTH0	N0	DXS0	DXE0				+CONT2
+CONT1	-4.5	0.2	15						+CONT2
+CONT2	X1	GROWTH1	N1	DXS1	DXE1				+CONT3
+CONT2	-1	1	20						+CONT3
+CONT3	X2	GROWTH2	N2	DXS2	DXE2				+CONT4
+CONT3	1			0.1	0.46				+CONT4

Field	Contents	Type	Default
ID	Unique bias number.	I > 0	Required
Xi	Begin coordinate of an interval. The interval ends at the next Xi entry.	R	Required
GROWTHi	GROWTHi is the ratio between the step size at the beginning of the interval and at the end of the interval. If it is smaller than 1.0 then the mesh refines when going from the beginning of the interval to the end of the interval. See Remarks 1 through 3.	R>0.0	See Remark 2.
Ni	Ni is the number of elements inside the interval.	I > 0	See Remark 2.
DXSi	DXSi is the start element size of the interval. See Remarks 1 through 3.	R>0.0	See Remark 2.
DXEi	DXEi is the end element size of the interval. See Remarks 1 through 3.	R>0.0	See Remark 2.

Remarks

1. The begin point of the first interval has to be equal the X0 field of the MESH entry and may be left unspecified. The end point of the last interval is given by X0+DX as specified by the MESH entry. In the example above the first interval is given by [-4.5, -1], The second one by [-1,1] and the last one by [1,4.5], assuming that X0+DX =4.5 on the MESH entry that uses the bias definition as a IBIDX.
2. To define the bias in an interval four fields are available. These are GROWTHi, Ni, DXSi, and DXEi. To specify the bias inside an interval, two of these four variables have to specified. The other two variables have to be left blank. In addition, the mesh size can be chosen constant by defining Ni and leaving GROWTHi, DXSi, and DXEi blank.

   This gives seven methods of specifying a bias;

   a. Define GROWTHi and Ni. DXSi and DXEi have to be left blank.
   b. Define DXSi and DXEi. GROWTHi and Ni have to be left blank.
   c. Define DXSi and Ni. GROWTHi and DXEi have to be left blank.
   d. Define DXEi and Ni. GROWTHi and DXSi have to be left blank.
   e. Define GROWTHi and DXSi. Ni and DXEi have to be left blank.
   f. Define GROWTHi and DXEi. Ni and DXSi have to be left blank.
   g. Only define Ni and leave GROWTHi, DXSi, an DXEi blank.

   For method b., it can happen that the biased elements do not exactly fit into the interval. To get a good fit, a small change to defined start and end step sizes is made. These changes in general amount to a few percents.

   For methods e. and f., a small change to the specified growth factor can be made so that the biased elements fit into the interval.
3. The algorithm for each method first determines an appropriate growth factor so that the biased elements fit into the interval. In addition the number of planes is determined. For methods b., c., and d., this is done by using bisection. The location of the planes is given by:

   $$X(1) = Xi$$

   $$\Delta X(1) = initial\ stepsize$$

   For $j = 2, N$

   $$X(j) = X(j-1) + \Delta X(j-1)$$

   $$\Delta X(j) = GROWFAC * \Delta X(j-1)$$

   with

   $$GROWTHi = GROWFAC^{N-1}$$

   Here, $X(j)$ denotes the location of the Euler plane, $\Delta X(j)$ denotes the step size between two adjacent planes and N denotes the number of Euler planes. The index j runs across the Euler planes The variable $GROWFAC$ denotes the grow factor between planes within the interval.

The locations of the planes $X(j)$ are written to the `.OUT` file as part of **MESH** output. In addition, the growth of the element sizes is written out in the next column. This is given as the ratio in element size between the layer of elements to the right of the plane and to the left of the plane. Let $X0$, $X1$, and $X2$ denote three subsequent planes, then the element size to the left of the x1-plane is given $X1 - X0$, and to the right, it is given by $X2 - X1$. The ratio by which the element size grows if one goes across the x1-plane is:

$$\frac{X2 - X1}{X1 - X0}$$

To get physically meaningful results, this value should not exceed 1.3 or be smaller than 0.7.

Within each interval, the ratio in element size equals $GROWFAC$. But the growth in element size between two adjacent elements that are in two different intervals can differ from $GROWFAC$. Here, each interval has a distinct $GROWFAC$ variable.

For both method b., e., and f., the growth factor $GROWFAC$ and start and end step sizes that are actually used can be obtained from this plane summary in the `.OUT` file. In this summary for each plane a growth factor and step size is specified. Also, the total number of elements is written out.

4. Here it is assumed that the **BIAS** ID-number was used on the `IBIDX` field of the **MESH** entry.

BJOIN

Breakable Join

Defines (multiple) pairs of grid points of one-dimensional and/or shell elements to be joined during the analysis. When the failure criterion for a grid point pair is satisfied, the grid point pair is removed from the join and the grid point motion is computed for the separate grid points. The join ceases to exist when all pairs of the join have failed, after which all of the grid points of the join are treated as separate grid points

Format and Example

1	2	3	4	5	6	7	8	9	10
BJOIN	BID	SID	TOL	TYPE	CRIT	VALUE1	VALUE2	VALUE3	+
BJOIN	1	2	1.E-04	COMPO	FORCE	1.E3	1.E4	1.E3	+
+	VALUE4	VALUE5	VALUE6	EQUIV	MULTI				+
+	1.E5	1.E4	1.E2	YES	YES				+
+	VALUE7	VALUE8	VALUE9						
+									

Field	Contents			Type	Default
BID	BJOIN number			I > 0	Required
SID	SET1 number			I > 0	Required
TOL	Tolerance used in matching grid-point pairs			R ≥ 0.0	1.E-4
TYPE	Type of failure criterion:			C	FOMO
	FOMO	Constant force and/or moment.			
		CRIT	No meaning (ignored).	C	Blank
		VALUE1	Force at failure.	R ≥ 0.0	1.E20
		VALUE2	Moment at failure.	R ≥ 0.0	1.E20
	USER	User-defined failure.			
		CRIT	No meaning (ignored).	C	Blank
		VALUE1	Name of the user-defined criterion to be used in the user subroutine.	C	Required
	TIME	Failure at a specified time.			
		CRIT	No meaning (ignored).	C	Blank
		VALUE1	Time of failure.	R ≥ 0.0	1.E20
	COMPO	Component failure at constant values.			
		CRIT	Criterion for failure.	C	BOTH

Field	Contents			Type	Default
		FORCE	Failure on forces.		
		MOMENT	Failure on moments.		
		BOTH	Failure on force and moment.	R ≥ 0.0	1.E20
		VALUE1	x-force at failure.	R ≥ 0.0	1.E20
		VALUE2	y-force at failure.	R ≥ 0.0	1.E20
		VALUE3	z-force at failure.	R ≥ 0.0	1.E20
		VALUE4	x-moment at failure.	R ≥ 0.0	1.E20
		VALUE5	y-Moment at failure.	R ≥ 0.0	1.E20
		VALUE6	z-Moment at failure.		
	SPOTWELD	Spot weld-type failure.			
		CRIT	No meaning.		
		VALUE1	Failure force in tension.	R ≥ 0.0	No failure
		VALUE2	Failure force in compression.	R ≥ 0.0	No failure
		VALUE3	Failure force in shear.	R ≥ 0.0	No failure
		VALUE4	Failure torque.	R ≥ 0.0	No failure
		VALUE5	Failure bending moment.	R ≥ 0.0	No failure
		VALUE6	Failure total force.	R ≥ 0.0	No failure
		VALUE7	Failure total moment.	R ≥ 0.0	No failure
		VALUE8	Failure time.	R ≥ 0.0	No failure
	RUPTURE	Rupture-like failure for 1-D grid points connected to shell grid points. Used to model skin-stringer delamination.			
		CRIT	No meaning.		
		VALUE1	Failure force per unit length in tension.	R ≥ 0.0	No failure
		VALUE2	Failure force per unit length in compression.	R ≥ 0.0	No failure
		VALUE3	Failure force per unit length in shear.	R ≥ 0.0	No failure
		VALUE4	Failure torque per unit length.	R ≥ 0.0	No failure
		VALUE5	Failure bending moment per unit length.	R ≥ 0.0	No failure
		VALUE6	Failure total force per unit length.	R ≥ 0.0	No failure
		VALUE7	Failure total moment per unit length.	R ≥ 0.0	No failure
		VALUE8	Failure time.	R > 0.0	No failure

Field	Contents				Type	Default
		VALUE9	Position of the stringer with respect to the skin element it is connected to:		C	MID
			MID	Stringer and the skin are at the same location.		
			UPPER	Stringer is located on the upper side of the skin.		
			LOWER	Stringer is located on the lower side of the skin		
EQUIV	Equivalence the positions of the grid points at time step zero.				C	YES
	YES	The positions of the two grid points are equivalenced as: $x_{bjoin} = \frac{1}{2}[x_{grid_1} + x_{grid_2}]$				
	NO	The positions of the two grid points are not equivalenced. The BJOIN behaves as a rigid body with the correct inertial properties until failure occurs.				
MULTI	Multiple breakable joins, where the grid points must be entered as a sequence of BJOIN pairs.				C	NO
	YES	The grid points are entered on the SET1 entry as a sequence of BJOIN pairs.				
	NO	Dytran creates BJOIN pairs for every two grid points entered on the SET1 entry when the grid point positions fall within the tolerance (TOL).				
	Independent of the setting of MULTI (either YES or NO), all BJOIN pairs that fall within the defined tolerance (TOL) are merged into one multiple breakable join					

Remarks

1. If the TYPE field is set to USER, the user subroutine must be present in the file referenced by the FMS statement.
2. The breakable joins can only be used for grid points of Lagrangian one-dimensional and shell elements. Note that any grid point can be made into a one-dimensional grid-point type by connecting a dummy spring to the grid point.

3. The constant force or constant moment failure criterion (TYPE=FOMO) is met once the following inequality is true:

$$(F_{x1}-F_{x2})^2+(F_{y1}-F_{y2})^2+(F_{z1}-F_{z2})^2>F^2$$

In the above formula, F is either a force or a moment. F_{max} is the value defined in the VALUEn fields.

4. If component failure is requested (TYPE=COMPO), the comparison is performed for each component of the force and moment vector. Depending on the criterion-type definition, the forces, the moments, or both are taken into account to determine whether the join fails.
5. In component failure, note that if one of the determining failure component values is left blank, this component can never cause the join to fail.
6. The first entity that satisfies the criterion for failure will cause the join to fail.
7. The undefined components in component failure are automatically set to infinity. This means that when failure on force components is requested, the moment criteria are set to infinity. The same is true for the forces when moment component failure is requested.
8. The user-defined criterion name can be a maximum of eight characters long.
9. At the moment of failure, an informational message is written to the output file.
10. The breakable joins for skin-stringer delamination (TYPE=RUPTURE) can only be used for beam-shell element connections. This type of connection can also be defined by the PWELD entry. The PWELD1 definition has an additional advantage in that it gives you access to the load on the connection. SeePWELD1 for more details.
11. A solid-shell connection, like for example the connection of the facing and core of a sandwich structure can be modeled using the PWELD2 entry. See PWELD2for more details.

BODYFOR

Body Force Loading

Defines a body force loading per unit mass.

Format and Example

1	2	3	4	5	6	7	8	9	10
BODYFOR	BID	TYPE	TYPEV						+
BODYFOR	100	EULER							+
+	CID	SCALE	VALUE	N1	N2	N3	NORMALIZE		
+	5	TABLE	13	1.	0.	0.			+
+									

Field	Contents		Type	Default
BID	Unique body force number		I > 0	Required
TYPE	Type of entity:		C	LAGRANGE
	LAGRANGE	Lagrangian type of grid point.		
	EULER	Eulerian type of element.		
	ELLIPS	Ellipsoid		
	GRID	List of Lagrangian grid points		
TYPEV	Name or ID of type of entity.		C or I	See Remark 1.
	TYPE	TYPEV		
	LAGRANGE	ELEM1D		
		SHTRIA		
		SHQUAD		
		MEMTRIA		
		LAGSOLID		
	EULER	EULHYDRO		
		EULSTREN		
		EULMM		
	ELLIPS	ELLIPS ID		
	GRID	SET1 ID		
CID	Number of a CORDxxx entry		I ≥ 0	0
SCALE	Scale factor for the load:		C	CONSTANT
	CONSTANT	Constant scale factor.		

Field	Contents		Type	Default
	TABLE	Tabular input for the scale factor.		
VALUE	Value or TABLE id for SCALE		I or R	Required
N1,N2,N3	Components of a vector giving the load (acceleration) direction. At least one must be nonzero.		R	See Remark 2.
NORMALIZE	Determine whether the direction vector (N1, N2, N3) is normalized. Values are ON and OFF.		C	ON See Remark 4.

Remarks

1. The default for entity TYPEV is all entities of TYPE.
2. By default the components are zero, but at least one of them should be nonzero.
3. Only one BODYFOR entry per type of entity TYPEV is allowed.
4. The bodyforce per unit mass is computed as:

   VALUE*(N1,N2,N3) or as Table value * (N1,N2,N3)

   By default, the vector (N1,N2,N3) will be normalized.

   For example, entering VALUE =9.8 and (N1,N2,N3) = (2,0,0) will give a bodyforce per unit mass of 9.8*(1,0,0)

   To avoid this normalization the field NORMALIZE has to be set to "OFF"

   Specifying VALUE =9.8, (N1,N2,N3) = (2,0,0) and NORMALIZE="OFF" will give 9.8*(2,0,0) = (19.6,0,0).

BODYFR1 Body Force for Eulerian Regions

Defines body force for Eulerian regions. The Eulerian regions are defined by geometric shapes. For each coordinate direction a time-depended acceleration can be defined.

Format and Example

1	2	3	4	5	6	7	8	9	10
BODYFR1	SID								+CONT1
BODYFR1	300								+CONT1
+CONT1	TYPE1	VID1	MID1	ACCX	ACCY	ACCz	LEVEL		+CONT2
+CONT1	BOX	400	100	100	200	300	0.0		+CONT2
+CONT2	TYPE2	VID2	MID2	ACCX	ACCY	ACCz	LEVEL		+CONT3
+CONT2	ELEM	500	200	400	500	500	1.0		+CONT3
+CONT3	TYPE3	VID3	MID3	ACCX	ACCY	ACCz	LEVEL		
+CONT3	CYLINDER	300	300	700	800	900	2.0		

Field	Contents		Type	Default
SID	Unique **BODYFR1** number referenced from a PEULER1 entry.		I > 0	Required
TYPEi	The type of Eulerian region.		C	Required
	SURF	Region inside or outside a multifaceted surface		
	SPHERE	Region inside a sphere.		
	CYLINDER	Region inside a cylinder.		
	BOX	Region inside a box.		
	ELEM	Region defined by element list.		
VIDi	Number of a geometric entity, a SET1 number, or number of a MATINI entry.		I > 0	Required
MIDi	Number of a DMAT entry to which the body force will be applied.		I > 0	Required
ACCi	Unique table number that defines the variation of acceleration in time. ACCX, ACCY, and ACCZ, respectively denote the acceleration in the x-,y- and z-direction.		I > 0	Required
LEVELi	Level indicator for this material and initial values.		R	0.0

Remarks

1. **BODYFR1** is only available for the multi-material Euler solver.
2. The combination of Eulerian region and material ID determines where the body force load is applied.
3. It is allowed to cover only part of the Euler domain with **BODYFR1** definitions.
4. All level indicators `LEVELi` must have different values. The level indicator can be negative.
5. To increase the accuracy of the region definition, the MICRO parameter can be used.
6. The total acceleration per element can be visualized by requesting the Eulerian scratch variable `FLUXVOL`.

BOX

Defines the Shape of a Box

Box shape used in the initial condition definition on the TICEUL entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
BOX	VID		X0	Y0	Z0	DX	DY	DZ	+CONT1
BOX	4		0.	0.	0.	1.	1.	1.	+CONT1

Field	Contents	Type	Default
VID	Unique box number.	I > 0	Required
X0, Y0, Z0	Coordinates of point of origin	R	Required
DX, DY, DZ	Width of box in different directions	R	Required

Remarks

1. The box is aligned with the coordinates axis.
2. Initial conditions are defined for the elements that are fully or partially inside the box. See *Dytran User's Guide*, Chapter 3: Constraints and Loading, Eulerian Loading and Constraints
3. See also TICEUL Bulk Data entry.

BOX1

Defines the Shape of a General BOX

Box1 shape used in the initial condition definition on the TICEUL entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
BOX1	VID								+
BOX1	4						.	.	+
+			X1	Y1	Z1	X2	Y2	Z2	+
+			0	0	0	0	0	1	+
+			X3	Y3	Z3	X4	Y4	Z4	+
+			0	0	1	1	0	0	+
+			X5	Y5	Z5	X6	Y6	Z6	+
+			0	1	0	0	0	1	+
+			X7	Y7	Z7	X8	Y8	Z8	
+			0	0	1	1	1	0	

Field	Contents	Type	Default
VID	Unique box1 number.	I > 0	Required
X1,..,Z8	Coordinates of 8 grid points	R	Required

Remarks

1. The Box1 allows a general box and edges do not need to be aligned with the coordinates axis. The eight grid points define the box identical to the CHEXA grid point numbering. Points may coincide as illustrated in the example values. These values give a pyramid shape.
2. Initial conditions are defined for the elements that are fully or partially inside the box. See Eulerian Loading and Constraints.
3. See also TICEUL Bulk Data entry.

CBAR

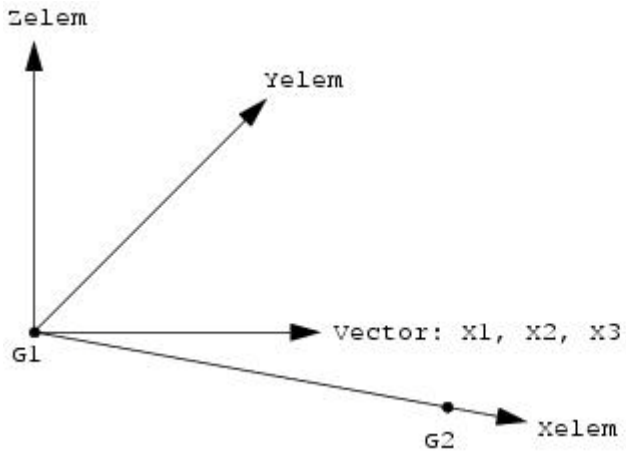
Bar Element Connection

Defines a simple beam element.

Format and Example

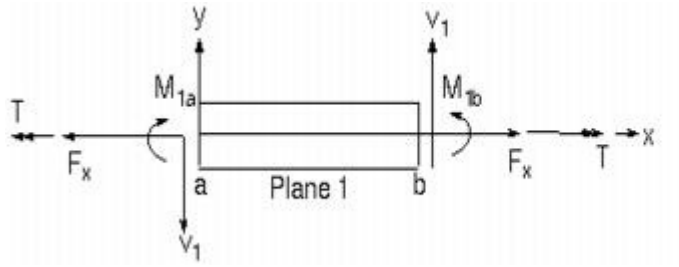
1	2	3	4	5	6	7	8	9	10
CBAR	EID	PID	G1	G2	X1, G3	X2	X3		
CBAR	2	39	7	3	3	13			

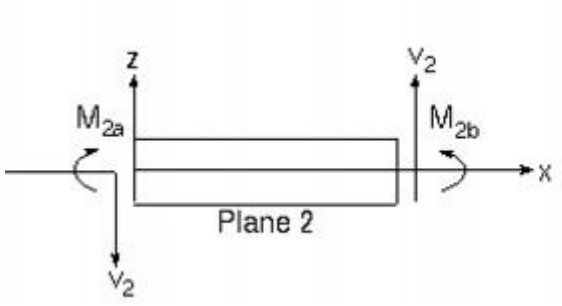
Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PBARor PBEAM property entry.	I > 0	EID
G1, G2	Grid-point numbers at the ends of the beam. G1 must not be the same as G2.	I > 0	Required
G3	Grid-point number to specify the vector defining the local x-y plane for the element. G3 must not be co-linear with G1 and G2.	I > 0	See Remark 2.
X1, X2, X3	Components of a vector at G1 in the basic coordinate system that lies in the element x-y plane.	R	See Remark 3.



Remarks

1. The element number must be unique with respect to all other element numbers.
2. The third grid point is used to specify a vector from `G1` to `G3`. The local x-axis of the beam is in the direction of the beam from point `G1` to `G2`. The local y-axis is perpendicular to the beam in the plane containing the vector from `G1` to `G3`. The local z-axis is perpendicular to the local x and y-axes (see *Dytran User's Guide*, Chapter 2: Elements, Beam Elements).
3. If field 6 (`X1`, `G3`) is an integer, `G3` is used to define the x-y plane. If field 6 (`X1`, `G3`) is real, `X1`, `X2`, and `X3` are used.
4. The following figures define the elemental force and moment sign convention (a and b are equivalent with G1 and G2, respectively).





CBEAM

Beam-Element Connectivity

Defines a beam element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CBEAM	EID	PID	G1	G2	X1,G3	X2	X3		+
CBEAM	2	39	7	3	13				+
+			W1A	W2A	W3A	W1B	W2B	W3B	+
+				3.0					+
+			COORD						
+									

Field	Contents		Type	Default
EID	Unique element number.		I > 0	Required
PID	Number of a PBEAM or PBEAM1 property entry.		I > 0	EID
G1, G2	Grid-point numbers at the ends of the beam. G1 must not be the same as G2.		I > 0	Required
G3	Grid-point number to specify the vector $\vec{v}$, defining the local x-y plane for the local element. G3 must not be co-linear with G1 and G2.		I > 0	See Remark 2.
X1, X2, X3	Components of a vector $\vec{v}$ at G1, in the basic coordinate system that lies in the element x-y plane.		R	See Remark 3.
W1A, W2A, W3A W1B, W2B, W3B	Components of offset vectors, measured in the displacement coordinate systems at grid points A and B, from the grid points to the end points of the axis of shear center.		R	0.0
COORD	Coordinate frame in which offset is defined.		C	GLOBAL
	GLOBAL	Vector is defined in Global coordinate System.		
	LOCAL	Vector is defined in Local Beam Coordinate System.		

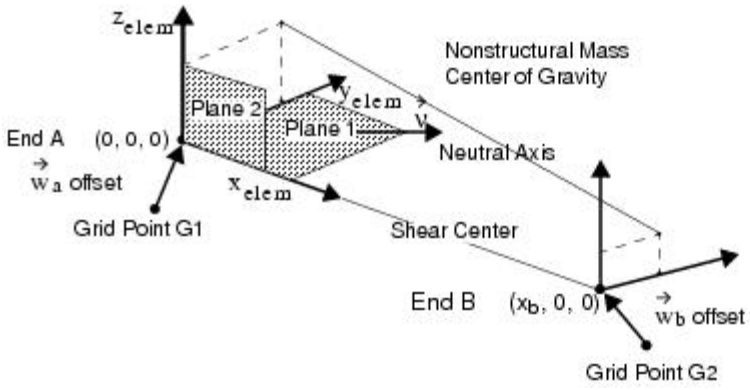


Figure 5-1 CBEAM Element Geometry System

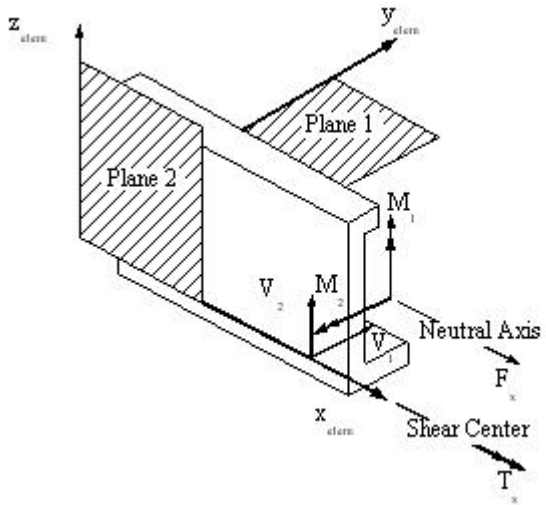


Figure 5-2 CBEAM Internal Element Forces and Moments.

Remarks

1. The element number must be unique with respect to all other element numbers.
2. The third grid point is used to specify a vector from `G1` to `G3`. The local x-axis of the beam is in the direction of the beam from point `G1` to `G2`. The local y-axis is perpendicular to the beam in the plane containing the vector from `G1` to `G3`. The local z-axis is perpendicular to the local x- and y-axes (See *Dytran User's Guide*, Chapter 2: Elements, Beam Elements).
3. If field 6 (`X1`, `G3`) is an integer, `G3` is used to define the x-y plane. If field 6 (`X1`, `G3`) is real, `X1`, `X2`, and `X3` are used.
4. Offset vectors are treated like rigid elements.

CBUSH

Generalized Spring-and-Damper Connection

Defines a generalized spring-and-damper structural element that may be nonlinear.

Format and Example

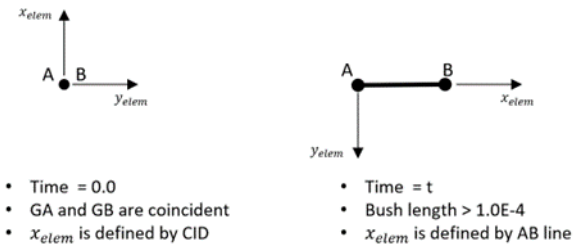
CBUSH	EID	PID	GA	GB	GO/X1	X2	X3	CID	+
CBUSH	3	1	1	100	75				
+	S								
+	0.5								

Field	Contents	Type	Default
EID	Element identification number.	I > 0	Required
PID	Property identification number of a PBUSH entry.	I > 0	Required
GA, GB	Grid point identification number of connection points.	I ≥ 0	See Remark 3.
GO	Grid-point number to specify the vector defining the local x-y plane of the element. GO must not be co-linear with GA and GB.	I > 0	See Remark 4.
Xi	Components of a vector at GA in the basic coordinate system that lies in the element x-y plane.	R	See Remark 4.
CID	Element coordinate system identification. A 0 value means the basic coordinate system will be used.	I ≥ 0	See Remark 5.
S	Location of spring-damper relative to GA.	R	0.5

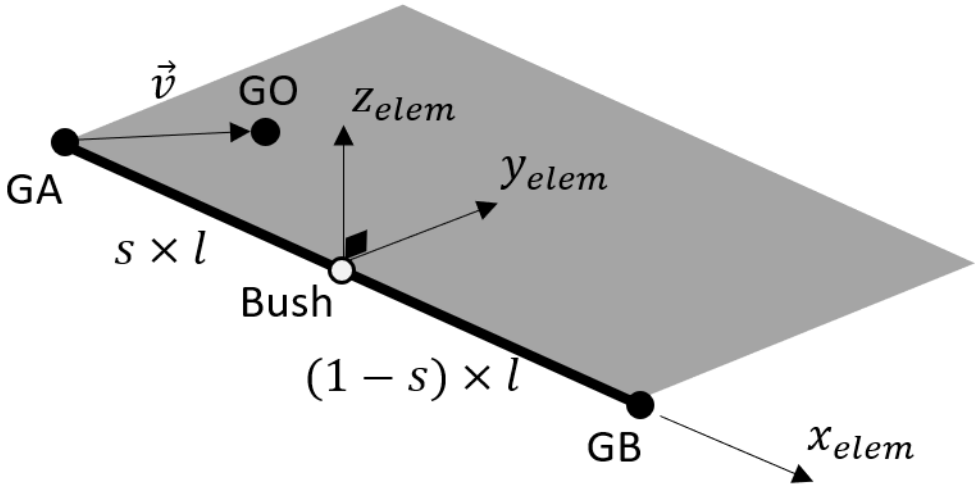
Remarks:

1. Element identification numbers should be unique with respect to all other element identification numbers.
2. The property entry PBUSH defines the Bush characteristics.
3. If either GA or GB (not both) is blank or zero, the element connected to the ground.
4. The third grid point, GO, or vector (X1, X2, X3) is used to specify a vector from GA to GO which lies in the element x-y plane. The local x-axis of the Bush is in the direction GA to GB. The local y-axis is perpendicular to the Bush x-axis in the plane containing the vector from GA to GO. The local z-axis is perpendicular to the local x and y-axes.

5. When GA and GB are coincident, CID must be assigned. CID defines the initial coordinate system of the Bush element. As soon as the element length which is under deformation becomes greater than 1.0E-4, the local x-axis will be in the direction of GA to GB. Also, the local y-axis will change accordingly, and consequently, the z-axis will be equal to the cross product of x-axis to y-axis. This may cause a sudden change in Bush direction, as shown in the following figure, which needs to be considered.



6. Bush element is placed on the x_elem axis between point A and B. Unlike Bush implementation in Nastran, Bush cannot be located out of AB line.

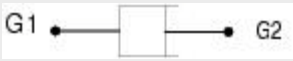


CDAMP1 Damper Connectivity

Defines a scalar damper element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CDAMP1	EID	PID	G1	C1	G2	C2			+
CDAMP1	19	6	7	3	104	3			+
+	CORD	FOLLOW							
+	3	G1							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PDAMP property entry.	I > 0	EID
G1,G2	Grid-point numbers at the ends of the damper. G1 must not be the same at G2. If either G1 or G2 are zero, the damper is connected to the ground.	I ≥ 0	0
C1,C2	Degree of freedom at G1 and G2 where the damper is connected.	1 ≤I ≤ 6	Required
			
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined.	I ≥ 0	0
FOLLOW	CORD: direction vector follows the motion of the coordinate system as specified under CORD. G1: direction vector follows the motion of end point G1. G2:direction vector follows the motion of end point G2.	C	CORD

Remarks

1. The element number must be unique with respect to all other element numbers.
2. The damper always acts in the direction given by C1 and C2 regardless of the relative positions of the grid points. CVISC defines a damper with an orientation that changes during the analysis.
3. Setting G1 or G2 to zero gives a grounded damper.
4. The damper can connect translational or rotational degrees of freedom.
5. The property entry PDAMP defines the damper characteristic.

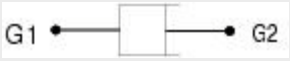
6. If the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.
7. The coordinate system CORD must be rectangular.
8. For fast rotating structures, it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.

CDAMP2 Linear Damper

Defines a linear scalar damper element where the damping coefficient is defined directly.

Format and Example

1	2	3	4	5	6	7	8	9	10
CDAMP2	EID	B	G1	C1	G2	C2			+
CDAMP2	19	2.4 E3	7	3	14	3			+
+	CORD	FOLLOW							
+	3	G1							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
B	Damping coefficient. (Force/velocity).	R	0.0
G1, G2	Grid-point numbers at the end of the damper. G1 must not be the same as G2. If either G1 or G2 are zero, the damper is connected to the ground.	I ≥ 0	0
C1, C2	Degree of freedom at G1 and G2 where the damper is connected.	1 ≤I ≤ 6	Required
			
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined.	I ≥ 0	0
FOLLOW	CORD direction vector follows the motion of the coordinate system as specified under CORD.	C	CORD
	G1: direction vector follows the motion of end point G1.		
	G2: direction vector follows the motion of end point G2.		

Remarks

1. The element number must be unique with respect to all other element numbers.
2. The damper always acts in the direction given by C1 and C2, regardless of the relative positions of the grid points. CVISC defines a damper with an orientation that can change during the analysis.
3. Setting G1 or G2 to zero gives a grounded damper.
4. The damper can connect translational or rotational degrees of freedom.
5. CDAMP1 can also be used to define linear scalar dampers. When there are many dampers with the same damping coefficient, it is more efficient to use CDAMP1.

6. When the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.
7. The coordinate system CORD must be rectangular.
8. For fast rotating structures, it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.

CELAS1

Scalar-Spring Connection

Defines a scalar-spring element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CELAS1	EID	PID	G1	C1	G2	C2			+
CELAS1	2	6	6	2	8	1			+
+	CORD	FOLLOW							
+	3	G1							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PELASn property entry.	I > 0	EID
G1, G2	Grid-point number.	I ≥ 0	0
C1, C2	Component number.	0 ≤ I ≤ 6	0
			
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined. It is used only when FOLLOW=CORD.	I ≥ 0	0
FOLLOW	CORD: direction vector follows the motion of the coordinate system as specified under CORD. G1: direction vector follows the rotational motion of end point G1. (See remark 7) G2: direction vector follows the rotational motion of end point G2. (See remark 7)	C	CORD

Remarks

1. A grounded spring is defined by setting G1 or G2 to zero in which case the corresponding C1 or C2 is zero or blank. (A grounded grid point is a grid point where the displacement is constrained to zero.)
2. Element numbers must be unique with respect to all other element numbers.
3. The connection grid points G1 and G2 must be distinct.
4. If the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.

5. The coordinate system CORD must be rectangular.
6. For fast rotating structures it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.
7. When FOLLOW = G1 or G2, the new direction vector is used and the nodal rotation of G1 and G2 is applied. Although G1 and G2 move in translational direction and the direction from G1 to G2 gets changed, yet the direction vector will not change without the nodal rotation at G1 and G2.

CELAS2 Scalar-Spring Property and Connection

Defines a scalar-spring element where the spring stiffness is defined directly.

Format and Example

1	2	3	4	5	6	7	8	9	10
CELAS2	EID	K	G1	C1	G2	C2			+
CELAS2	28	6.2+3	32	1	19	4			+
+	CORD	FOLLOW							
+	3	G1							

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
K	The stiffness of the scalar spring	R	0.
G1, G2	Grid-point number	I ≥ 0	0
C1, C2	Component number	0 ≤ I ≤ 6	0
			
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined. It is used only when FOLLOW=CORD.	I ≥ 0	0
FOLLOW	CORD: direction vector follows the motion of the coordinate system as specified under CORD. G1: direction vector follows the rotational motion of end point G1. (See remark 9) G2: direction vector follows the rotational motion of end point G2. (See remark 9)	C	CORD

Remarks

1. A grounded spring is defined by:
   a. Setting G1 or G2 to zero in which case the corresponding C1 or C2 is zero or blank.
   b. Using a scalar point for G1 and/or G2 in which case the corresponding C1 and/or C2 is zero or blank. (A grounded grid point is a grid point where the displacement is constrained to zero.)
2. Element numbers must be unique with respect to all other element numbers.
3. This entry completely defines the element since no material or geometric properties are required.
4. The two connection points G1 and G2 must be distinct.

5. If the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.
6. The coordinate system CORD must be rectangular.
7. For fast rotating structures it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.
8. If possible, use of the PELAS, CELAS1 entries is preferable. Many CELAS2 elements result in excessive input manipulation and storage.
9. When FOLLOW = G1 or G2, the new direction vector is used and the nodal rotation of G1 and G2 is applied. Although G1 and G2 move in translational direction and the direction from G1 to G2 gets changed, yet the direction vector will not change without the nodal rotation at G1 and G2.

CFACE

Face of an Element

Defines a face on an Eulerian or a Lagrangian element.

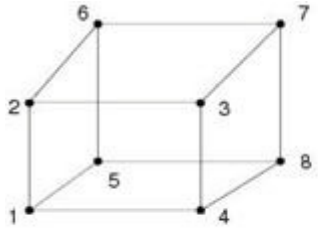
Format and Example

1	2	3	4	5	6	7	8	9	10
CFACE	FID	SID	EID	FACE					
CFAC4E	37	100	1796	4					

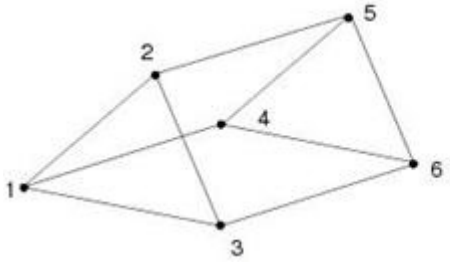
Field	Contents	Type	Default
FID	Unique face number.	I > 0	Required
SID	Number of a set of faces to which the face belongs. It is referenced by a FLOW or SURFACE entry.	I > 0	Required
EID	Element number to which the face is attached.	I > 0	Required
FACE	The number of the element face that is to be used. See Remark 3.	1 ≤ I ≤ 6	Required

Remarks

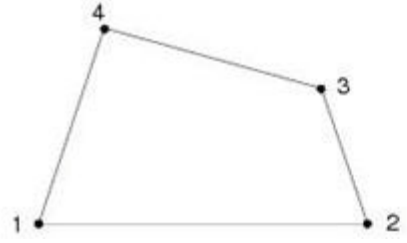
1. The face number FID must be unique with respect to all other face numbers.
2. The CSEG entry is also used to define faces in terms of the grid-point numbers. The CFACE1 entry is also used to define faces.
3. A negative face number indicates that the face normal direction is reversed.
4. The element-face numbers are as follows:



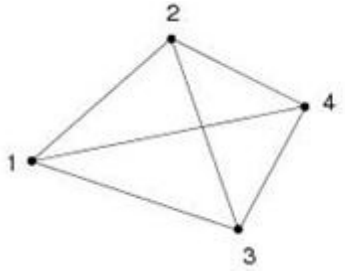
Face Number	Grid Points
1	1432
2	1265
3	1584
4	7856
5	7348
6	7623



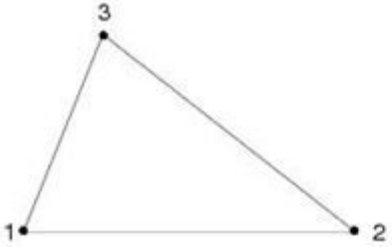
Face Number	Grid Points
1	132
2	1254
3	1463
4	645
5	6523



Face Number	Grid Points
1	1432
4	1234



Face Number	Grid Points
1	132
2	124
3	143
4	423



Face Number	Grid Points
1	132
4	123

CFACE1

Face of an Element

Defines a face on an element in terms of the element number and two grid points on the required face. This is particularly suitable for defining the faces on solid elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
CFACE1	FID	SID	EID	G1	G3/G4				
CFACE1	497	100	2796	32	4162				

Field	Contents	Type	Default
FID	Unique face number.	I > 0	Required
SID	Number of a set of faces to which the face belongs. It is referenced by a FLOW or a SURFACE entry.	I > 0	Required
EID	Element number to which the face is attached.	I > 0	Required
G1	Number of a grid point connected to a corner of the face.	I > 0	Required
G3	Number of a grid point connected to a corner diagonally opposite to G1 on the same face of a CHEXA or CPENTAelement. This applies to quadrilateral faces of CPENTA elements only. G3 must be omitted for a triangular face on a CPENTA element.	I > 0	Blank
G4	Number of the grid point of a CTETRA element that is not on the required face.	I > 0	Required

Remark

A PLOAD4 entry with an absolute pressure of 9999. is automatically converted to a CFACE1 entry. This makes defining CFACE1 entries in preprocessors very easy. See also *Dytran User's Guide*, Chapter 9: Running the Analysis, Using a Modeling Program with Dytran.

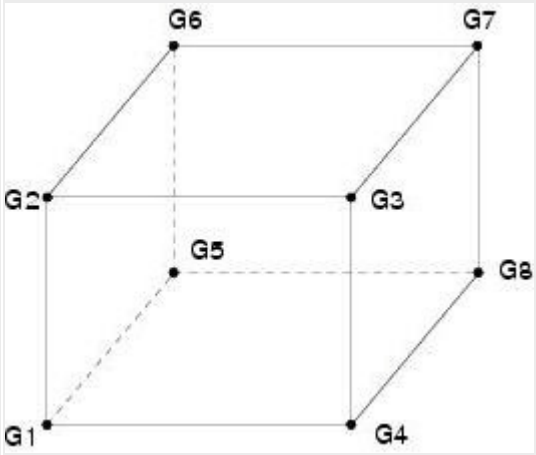
CHEXA

Element with Eight Grid Points

Defines an Eulerian or a Lagrangian element with eight corner grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
CHEXA	EID	PID	G1	G2	G3	G4	G5	G6	+
CHEXA	71	4	3	4	5	6	7	8	+
+	G7	G8							
+	9	10							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PSOLID or PEULER property entry.	I > 0	EID
G1-G8	Grid-point numbers of the connected grid points. They must all be unique. 	I > 0	Required

Remarks

1. The element number must be unique with respect to all other element numbers.
2. Grid points G1 through G4 must be given in consecutive order about one quadrilateral face. G5 through G8 must be on the opposite face with G5 opposite G1, G6 opposite G2, etc.
3. Number according to the figure shown in this CHEXA entry description.
4. The property number references a PSOLID or a PEULER entry. This determines whether the element is Lagrangian or Eulerian.
5. Only the first eight grid points on a CHEXA are used in Dytran. The excess is ignored.

CIFHEX Solid Interface Cohesive Zone Modeling Element

Linear three-dimensional interface element used to simulate the onset and progress of delamination.

Format and Example

1	2	3	4	5	6	7	8	9	10
CIFHEX	EID	PID	G1	G2	G3	G4	G5	G6	+
CIFHEX	700	701	101	102	103	104	105	106	+
+	G7	G8							
+	107	108							

Field	Contents	Type	Default
EID	Element identification number.	I > 0	Required
PID	Property number of a PCOHE entry.	I > 0	Required
G1-G8	Identification number of connected corner grid points. Required data for all eight corner grid points.	I > 0	Required

Remarks:

1. Element identification numbers should be unique with respect to all other element ID's of any kind.
2. The corresponding deformations are relative displacements between the top and bottom edge of the element. Normal direction of element is set from bottom surface to top surface.
3. The element is allowed to be infinitesimally thin; in which case edges defined by grids G1-G4 and G5-G8 may coincide. However, quadrilateral made by G1~G4 and G5~G8 must have proper area. If not, the results will not be accurate.

CMARKB2 Two-noded Marker Connectivity Definition

Defines a two-noded marker beam element by means of connecting two grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
CMARKB2	ID	PID	G1	G2					
CMARKB2	7	1	9	10					

Field	Contents	Type	Default
ID	Unique element number.	I > 0	Required
PID	Property ID referring to a PMARKER entry	I > 0	Required
G1	Grid-point number connectivity 1.	I > 0	Required
G2	Grid-point number connectivity 2.	I > 0	Required

Remarks

1. A CMARKB2 element may refer to two types of grid points:
   a. Structural grid points, thus grid points that are part of the connectivity of an element.
   b. Free grid points in space. These grid points do not have mass associated with them. The motion of these grid points is specified by the PMARKER property.
2. The ID must be unique in the model and may not be used as structural element ID.

CMARKN1 One-noded Marker Connectivity Definition

Defines a one-noded marker element on a grid point.

Format and Example

1	2	3	4	5	6	7	8	9	10
CMARKN1	ID	PID	G						
CMARKN1	7	1	9						

Field	Contents	Type	Default
ID	Unique element number.	I > 0	Required
PID	Property ID referring to a PMARKER entry	I > 0	Required
G	Grid-point number.	I > 0	Required

1. A CMARKN1 element may refer to two types of grid points:
   a. Structural grid points, thus grid points that are part of the connectivity of an element. In this case, the PMARKER ID is ignored.
   b. Free grid points in space. These grid points do not have mass associated with them. The motion of these grid points is specified by the PMARKER property.
2. The ID must be unique in the model and may not be used as structural element ID.

COHFRIC Cohesive Friction

Allows friction and sticking during tensile conditions at the coupling surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
COHFRIC	CID	MAXSTRS	FRIC	REFVEL					
COHFRIC	112	8E+10	8e+5	2					

Field	Contents	Type	Default
CID	Unique number of a COHFRIC entry.	I > 0.0	
MAXSTRS	Maximal normal stress. Allows tensile stresses at the coupling surface as long as the normal stress does not exceed MAXSTRS.	R ≥ 0.0	
FRIC	Friction stress under tensile conditions	R ≥ 0.0	
REFVEL	Reference value for velocity	R ≥ 0.0	

Remarks

1. If the cohesive friction parameters is uniform across the coupling surface then PARAM, COHESION can be used instead.
2. During tension any relative tangential velocity between coupling surface and Eulerian material will yield a shear stress whose magnitude equals $Fric \times min(1, \frac{V_{REL \cdot tangential}}{REFVEL})$. This is a viscous-like friction law.
3. This shear force opposes the relative tangential movement along the coupling surface.

CONM2 Concentrated Grid Point Mass and/or Inertia

Defines a concentrated grid point mass and/or inertia for Lagrangian elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
CONM2	ID	G		M				I	
CONM2	7	9		.1				4.4E-3	

Field	Contents	Type	Default
ID	Unique CONM2 number	I > 0	Required
G	Grid-point number	I > 0	Required
M	Mass	R	0.0
I	Inertia	R	0.0

Remarks

1. All grid points in the model must have mass associated with them, either by the properties of the elements attached to the grid point or by using a CONM2 entry.
2. When PARAM,CONM2OUT is set to NO, there is no summary on the CONM2 entries defined. This means that the mass, momentum, and energy of the CONM2s are not added to the material and cycle summaries. Setting PARAM,CONM2OUT,NO saves memory and CPU time.
3. The CONM2 results cannot be output on time-history or archive files.

CONTACT

Contact Surface

Defines contact between Lagrangian grid points and elements. The algorithm is based on contact between primary contact faces and secondary contact grid points.

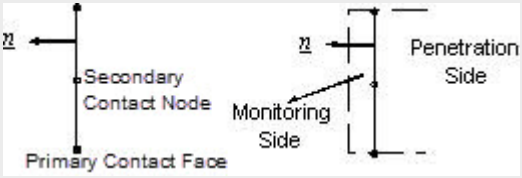
Format and Example

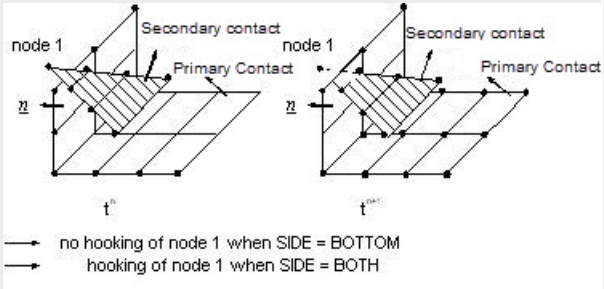
1	2	3	4	5	6	7	8	9	10
CONTACT	CID	STYPE	PTYPE	SID	PID	FS	FK	EXP	+
CONTACT	7	SURF	SURF	3	7	0.0	0.0	0.0	+
+	VERSION	SIDE	SEARCH	ADAPT	THICK	GAP	DAMPING	WEIGHT	+
+	V4	BOTH	FULL	NO	1.0	0.0	YES	BOTH	+
+			PEN	PENV	MONVEL	FACT	MONDIS	MONDISV	+
+			DISTANCE	1.E20	1.1	0.1	FACTOR	2.0	+
+	TSTART	TEND	REVERSE	INITPEN	PENTOL	INIID	INITMON	SECACT	+
+	0.0	1.E20	ON	ON	1.E20				+
+	DRWBEADF	CONTFORC	TOLPROJ1	TOLPROJ2	EVIEW				+
+									+
+	TENDNEW								
+									

Field	Contents		Type	Default
CID	Unique contact number.		I > 0	Required
STYPE	Type of entity used to define the secondary contact nodes:		C	SURF
	SURF	All nodes belonging to a SURFACE.		
	ELEM	All nodes belonging to a list of elements.		
	PROP	All nodes belonging to elements with certain property numbers.		
	MAT	All nodes belonging to elements with certain material numbers.		
	GRID	A list of grid points.		
	ELLIPS	A list of ellipsoid IDs.		See Remark 2.
	SURFEDGE	All edges belonging to a surface		See Remark 6.

Field	Contents		Type	Default
	DRAWBD	DRAWBD ID. It is only available when VERSION=DRAWBDV4.		
	DRWDSET	DRAWBD1 CSID. It is only available when VERSION=DRAWBDV4.		
PTYPE	Type of entity used to define the primary contact faces:		C	Blank
	blank	All faces belonging to the secondary contact SURFACE, or the faces belonging to the elements referenced by STYPE,SID. This option is only allowed for STYPE = SURF, ELEM, PROP, or MAT. The option results in a so-called single surface contact.		
	SURF	All faces belonging to a surface.		
	ELEM	All faces belonging to a list of elements.		
	PROP	All faces belonging to elements with certain property IDs.		
	MAT	All faces belonging to elements with certain material IDs.		
	ELLIPS	A list of ellipsoid IDs.		See Remark 2.
	SURFEDGE	All edges belonging to a surface.		See Remark 6.
SID	Number of a SURFACE entry if STYPE = SURF, or number of a SET1 entry if STYPE = ELEM, PROP, MAT, or GRID, or number of a SETC entry if STYPE = ELLIPS.		I > 0	Required
PID	Number of a SURFACE entry if PTYPE = SURF, or number of a SET1 entry if PTYPE = ELEM, PROP, MAT, or GRID, or number of a SETC entry if PTYPE = ELLIPS.		I > 0	
FS	Static coefficient of friction (See Remark 3.)		R	0.0
FK	Kinetic coefficient of friction (See Remark 3.)		R	0.0
EXP	Exponential decay coefficient (See Remark 3.)		R	0.0
VERSION	Version of the algorithm (See Remark 3.):		C	Required
	V2	Obsolete (use V4 version)		
	V3	Obsolete (use V4 version).		

Field	Contents			Type	Default
	V4	Most recent algorithm.			
		An option is available to assign spring/damper characteristics to the contact force (CONTFORC).			
	BELT	Suited for modeling contact between a belt element and a rigid structure.			
		Primary-secondary contact only.			
		The contact logic doesn't apply a contact force, but applies an enforced displacement and velocity that keeps the secondary contact nodes exactly on top of the primary contact face.			
		The secondary contact node does not slide relative to the primary contact face when the friction coefficient (FS) is set to 1E20.			
	BELT1	Identical to BELT algorithm, except that the secondary contact nodes are initially repositioned on top of the closest primary contact face.			
		All secondary contact nodes initially penetrated or within a distance of INITMON from a primary contact face, are repositioned.			
	DRAWBEAD	Suited for modeling a drawbead. STYPE Must be GRID. The list of secondary contact nodes must be ordered along the drawbead line. PTYPE Must be SURF/ELEM/MAT/PROP. The restraining force per unit drawbead length is specified in the field DRWBEADF.			
		STYPE	Must be GRID. The list of secondary contact nodes must be ordered along the drawbead line.		

Field	Contents			Type	Default
		PTYPE	Must be SURF. The restraining force per unit drawbead length is specified in the field DRWBEADF.		
	DRAWBDV4	Suited for modeling a drawbead in V4 contact. STYPE Must be GRID, DRAWBD, or DRAWBD1. PTYPE Must be SURF/ELEM/MAT/PROP. The restraining force per unit drawbead length is specified in the field DRWBEADF or in DRAWBD or DRAWBD1 definition.			
SIDE	Defines which side will be the monitoring side of a primary contact face. The opposite side of the primary contact face will be the penetration side.			C	BOTH
	BOTH	The side from which a secondary contact node approaches the primary contact face will become the monitoring side.			
	TOP	The monitoring side will always be on the side of the primary contact face that its normal is pointing at.			
	BOTTOM	The monitoring side is always on the opposite side of the primary contact face that its normal is pointing at.			
	The options TOP/BOTTOM are useful in the following cases:				
	1. When a secondary contact node initially is located on the primary contact face (see the picture below), the contact situation is uniquely defined, only if the TOP or BOTTOM side of the primary contact surface is defined.				
					

Field	Contents		Type	Default
	2. When hooking of secondary contact nodes on the wrong side of a primary contact face might occur. This often is the case when the primary contact face is at the edge of a shell element structure:  			
SEARCH	Defines the type of search algorithm:		C	FULL
	FULL	Regular search algorithm.		
	SLIDE	Special option for in-plane folded air bags. This option should be used with care in other applications.		
	BPLANE	Search algorithm, where a back-up plane region alleviates the problem for SLIDE when a secondary contact node has entered a dead region.		
	BPFULL	Search algorithm that combines features of the BPLANE and the FULL search contact algorithms.		
	See Remark 5. for a more detailed description of these methods.			
ADAPT	Defines whether the primary contact faces are (de)-activated based on element failure. Secondary contact nodes only check for contact with active primary contact faces:		C	NO
	NO	The contact is nonadaptive, and all the primary contact faces are active during the whole analysis.		
	YES	The contact is adaptive. SURF, ELLIPS and SURFEDGE types are not allowed. The primary contact faces are (de)activated according to the following logic:		

Field	Contents		Type	Default
		Shell elements – At time zero all the primary contact faces are active. Once an element fails, its corresponding primary contact face will be deactivated. The contact treats it as an actual hole.		
		Lagrangian solids – At time zero only the free faces are active. All the internal faces are deactivated. When an element fails, some of its faces might become free faces. These free faces will be activated. Once all the elements connected to a primary contact face have failed, it is deactivated for the remainder of the analyses. This logic allows for modeling of impact-penetration phenomena, and is sometimes called "eroding contact."		
	Note: (De)-activation of secondary contact nodes is selected on the SECACT field.			
THICK	Shell thickness scale factor. See Remark 2. and 4.		R	Required for VERSION = V4
GAP	Artificial contact thickness. See Remark 4.		R	0.0
DAMPING	[YES/NO] - VERSION V4 only. Specifies if a high frequency damping is active or not. The damping force is based on the relative velocity of a secondary contact node with respect to a primary contact face. The damping is preferably turned on in all cases, except for RIGID-RIGID contact. In RIGID-RIGID contact it can result in a substantial loss of energy.		C	YES
WEIGHT	For contact versions V2 and V4, the contact force is multiplied by a mass-weighting factor. The following options are available:		C	BOTH
	BOTH	$M_{scale} = \frac{M_{primary} M_{secondary}}{M_{primary} + M_{secondary}}$		
	SECNDRY	$M_{scale} = M_{secondary}$		
	PRIMARY	$M_{scale} = M_{primary}$		

<table>
<tr><th>Field</th><th>Contents</th><th></th><th>Type</th><th>Default</th></tr>
<tr><td></td><td>NONE</td><td>$M_{scale} = 1.0$</td><td></td><td></td></tr>
<tr><td></td><td>Recommended usage:
<table><tr><th>Secondary Contact Surface</th><th>Primary Contact Surface</th><th>Weight</th></tr><tr><td>Deformable</td><td>Deformable</td><td>BOTH</td></tr><tr><td>Deformable</td><td>Rigid</td><td>SECNDRY</td></tr><tr><td>Rigid</td><td>Deformable</td><td>PRIMARY</td></tr><tr><td>Rigid</td><td>Rigid</td><td>NONE</td></tr></table></td><td></td><td></td></tr>
<tr><td></td><td>Default setting, if ELLIPS is used in either STYPE or PTYPE entry, is as follows:
<table><tr><th>Secondary Contact Surface</th><th>Primary Contact Surface</th><th>Weight</th></tr><tr><td>Non-ELLIPS</td><td>ELLIPS</td><td>SECNDRY</td></tr><tr><td>ELLIPS</td><td>Non-ELLIPS</td><td>NONE</td></tr></table></td><td></td><td></td></tr>
<tr><td>PEN</td><td>Allowed penetration check.</td><td></td><td>C</td><td>No check</td></tr>
<tr><td></td><td>If the penetration depth exceeds a certain value it is assumed that the secondary contact node is in a bad contact state. No contact force is applied and the secondary contact node is taken out of the contact for the remainder of the calculation. This option is useful in the following applications:</td><td></td><td></td><td></td></tr>
<tr><td></td><td>1. In airbag analysis to prevent “locking” of the unfolding bag.</td><td></td><td></td><td></td></tr>
<tr><td></td><td>2. In crash analysis to prevent high contact forces in extremely folded regions that would require a much finer mesh without this option.</td><td></td><td></td><td></td></tr>
<tr><td></td><td>DISTANCE</td><td>The allowed penetration depth is specified in PENV.</td><td></td><td></td></tr>
<tr><td></td><td>FACTOR</td><td>The allowed penetration depth is equal to a factor times a characteristic length of the primary contact faces. The factor is specified in PENV.</td><td></td><td></td></tr>
<tr><td>PENV</td><td>Value of the allowed penetration depth or value of the FACTOR to calculate the allowed penetration depth.</td><td></td><td>R</td><td>No check</td></tr>
</table>

Field	Contents		Type	Default
MONVEL	The contact monitoring distance is increased by a value based on the relative velocity of a face of a primary contact and the node of a secondary contact. The increase is only used if the secondary contact node is moving towards the primary contact face, and is equal to: `MONVEL * (relative velocity) * DT`		R	1.1
FACT	Scale factor for the contact forces.		R > 0.0	0.1
	The default value for FACT works in most applications. When the secondary contact nodes penetrate too much, the contact can be made stiffer by increasing the value of FACT.			
	It is advised to limit the value of FACT to:			
		Single Surface Contact: `FACT = 0.5`		
		Primary-secondary Contact: `FACT = 1.0`		
	When a CONTFORC entry is defined for this contact, the value of FACT is not used. The contact force is based solely on the spring/damper characteristics as specified on the CONTFORC entry.			
MONDIS	Defines the fixed part of the monitoring distance.		C	FACTOR
	When the normal distance of a face of a primary contact and the node of a secondary contact becomes smaller than the monitoring distance the secondary contact node will tag itself to the primary contact face. The side from which the secondary contact node is moving towards the primary contact face becomes the monitoring region.			
	The monitoring distance contains a fixed part and a velocity dependent part. See MONVEL for a description of the velocity dependent part.			
	DISTANCE	The monitoring distance is specified in MONDISV.		
	FACTOR	The monitoring distance is equal to a factor times a characteristic length of the primary contact faces. The factor is specified in MONDISV.		
MONDISV	Value of the monitoring distance or value of the FACTOR to calculate the monitoring distance.		R	2.0

Field	Contents			Type	Default
TSTART	Time at which the contact is activated. This overrules a possible definition on an ACTIVE entry.			R ≥ 0	0.0
TEND	Time at which the contact is deactivated. This overrules a possible definition on an ACTIVE entry.			R ≥ 0	ENDTIME
REVERSE	[ON/OFF]			C	ON
	Automatic reversal of primary contact faces such that their normal point in the same direction. (See Remark 7.)				
INITPEN	[ON/OFF]			C	ON
	Allowed initial penetration check.				
	Each secondary contact node is checked for an initial penetration, and if the initial penetration depth is within an allowed limit.				
	If an initial penetration occurs, and the penetration depth falls within the allowed limit, a warning is issued.				
	If an initial penetration occurs and the initial penetration depth is larger than the allowed value, the contact between the face of a primary contact and the node of a secondary contact is not initialized. No warning is issued.				
PENTOL	Tolerance for the allowed initial penetration check.			R	1.E20
INIID	ID of a set of CONTINI entries used to define the initial contact state. This option is useful for in-plane folded air bags.			I > 0	Blank
INITMON	Fixed part of the monitoring distance used during the initialization. If not specified, the value of MONDIS is used.			R > 0.0	MONDIS
SECACT	Defines the method used to (de)activate the secondary contact nodes.			C	See Remark 9.
	VERSION=V2	Applies only when ADAPT = YES.			
		NO	The secondary contact nodes are deactivated after all its connected elements have failed.		
		YES	The secondary contact nodes are always active.		
	VERSION=V4	Applies for both ADAPT = YES and ADAPT = NO.			
	METHOD1	Applies to all V4 contacts.			
	METHOD2	Applies to all V4 contacts.			

Field	Contents	Type	Default
	METHOD3 Applies to all V4 contacts.		
	METHOD4 Applies to all V4 contacts.		
	METHOD1A Applies to primary-secondary V4 contacts only.		
	METHOD2A Applies to primary-secondary V4 contacts only.		
	METHOD3A Applies to primary-secondary V4 contacts only.		
	METHOD4A Applies to primary-secondary V4 contacts only		
	See Remark 9. for a detailed description of these methods.		
DRWBEADF	Drawbead force per unit length. If DRWBDRF in DRAWBD is defined, this will be ignored and instead DRWBDRF will be used.	R > 0.0	Required for VERSION = DRAWBEAD or DRAWBDV4.
CONTFORC	ID of a CONTFORC entry.	I > 0	Blank
	When specified, the contact force is not based on the Lagrangian multiplier method, but determined by spring/damper characteristics. The spring/damper characteristics are specified on a CONTFORC entry.		
	When the CONTFORC entry is specified, the value of FACT and DAMPING are not used.		
TOLPROJ1, TOLPROJ2	Projection tolerance for inside and outside corners. Faces are automatically extended to cover the "dead region" at corners. (See Remark 5.)	R > 0.0	1.E-3
EVIEW	The view angle (in degrees) of edges. Used only in case of edge-to-edge contact.	0 < R < 90	See Remark 6.
TENDNEW	Deactivation time for new contact search algorithm for BPFULL contact.	R > 0.0	1.E20 See Remark 8..

Remarks

1. See also the information on the contact algorithm in the *Getting Started Manual*.
2. The SETC ID referred to by ELLIPS may contain more than one ellipsoid. The ELLIPS option may only use the V4 contact algorithm and the default thickness factor for the ELLIPS option is 0.0.

3. The coefficient of friction is given by:

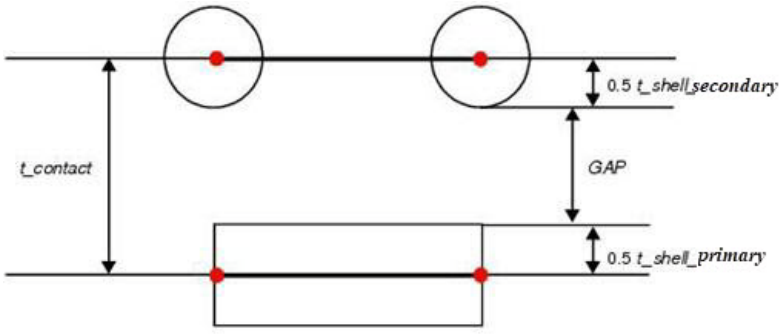
$$\mu = \mu_k + (u_s - \mu_k)e^{-\beta v}$$

where

μ_s	=	static coefficient of friction FS
μ_k	=	kinetic coefficient of friction FK
β	=	exponential decay coefficient EXP
v	=	relative sliding velocity at the point of contact

4. When a nonzero value has been specified for `THICK` and/or `GAP`, a contact thickness is assigned to both the face of a primary contact and the node of a secondary contact:

CASE 1: Shell-to-shell Contact



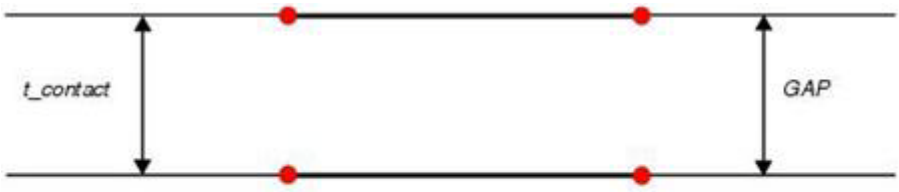
The contact thickness is equal to:

$t_contact = THICK$ * 0.5 * t_shell_secondary + $THICK$ * 0.5 * t_shell_primary + GAP

where

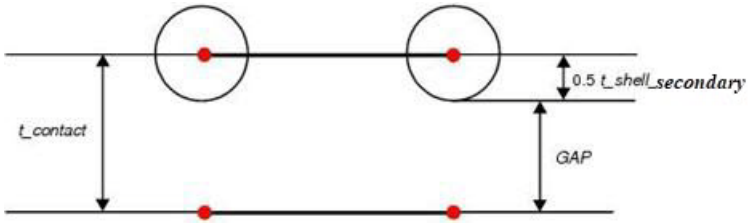
$t_contact$	=	contact thickness
$THICK$	=	scale factor for shell thickness
t_shell	=	shell thickness
GAP	=	artificial contact thickness

CASE II: Solid-to-solid Contact



t_contact = *GAP*

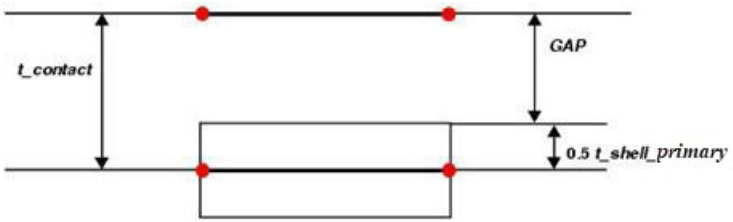
CASE III: Shell-to-solid Contact:



The contact thickness is equal to:

:_contact = *THICK* * 0.5 * t_shell_secondary + *GAP*

CASE IV: Solid-to-shell Contact:



The contact thickness is equal to:

t_contact = *THICK* * 0.5 * t_shell_primary + *GAP*

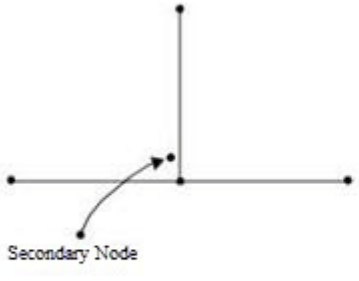
The shell thickness is zero for the faces of Lagrangian solids.

The shell thickness for secondary contact nodes is not calculated for `STYPE = GRID`.

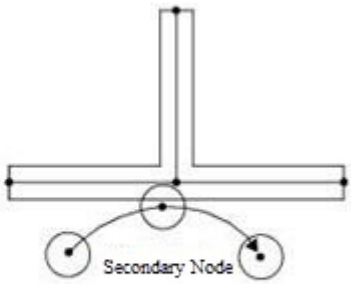
For all other options, the shell thickness of each secondary contact node is calculated as the average thickness of those elements in the secondary contact surface that are connected to the secondary contact node.

Contact occurs when the contact thickness of a secondary contact node overlaps the contact thickness of a primary contact face. This is the best physical contact representation of shell structures. There are also several other advantages to using a contact thickness:

a. Use of a contact thickness prevents "hooking" in case of T-joints:

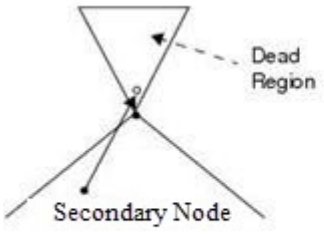


With no contact thickness, the slave node slightly penetrates and hooks on the vertical face.

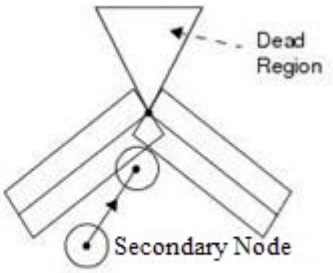


With contact thickness, the slave node never penetrates and hooking does not occur.

b. Use of a contact thickness prevents losing contacts in the "dead region" on the "penetrated side" of neighboring primary contact faces. When a secondary contact node enters the "dead region" between neighboring primary contact faces, it is not projected on either face, and the contact is lost:

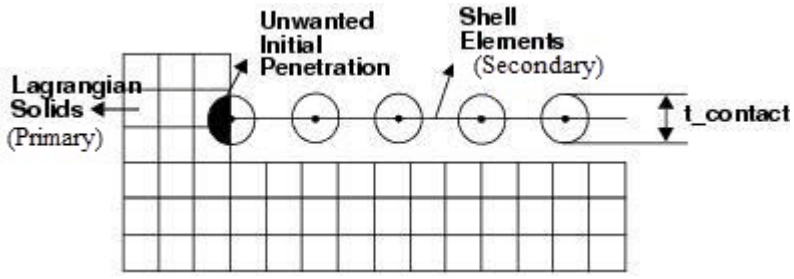


With No Contact Thickness



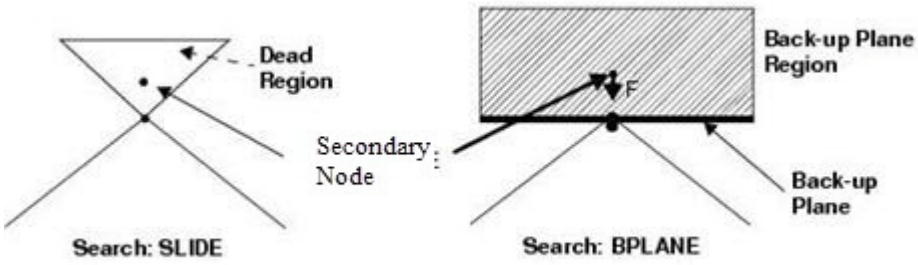
With Contact Thickness

Using a contact thickness has the disadvantage that an unwanted initial penetration might occurs where the edge of shell elements meets a primary contact surface. The following is a good example:



5. The search for the closest primary contact face for a secondary contact node is performed by a normal projection of the secondary contact node on the faces of the primary contact surface. For the search option FULL, all faces of the primary contact surface are taken into account. This is the most reliable option, but takes the most computational time.

   The SLIDE search option searches for the closest primary contact face under the assumption that a secondary contact node will only slide from the current primary contact face to its neighbors during one time step. This search algorithm is much faster than the FULL option. However, this search option can cause problems for secondary contact nodes that have entered the "dead-region" on the "penetrated side" between neighboring primary contact faces, where it can not be projected on either face, and the contact is lost because of that.



   The BPLANE search option alleviates the problems for secondary contact nodes that have entered the "dead-region" by creating automatically a plane perpendicular to the folding line between two primary contact faces. Contact for secondary contact nodes that reside in the back-up plane area is preserved and the nodes remain in the contact search algorithm. A force perpendicular to the BPLANE is applied to those nodes, since they are on the penetrated side of the primary contact surface.

   Because the BPLANE search algorithm is very fast, it is the recommended contact search algorithm for air bag analysis.

   The BPFULL algorithm combines features of the FULL search contact algorithm and the BPLANE search contact algorithm. The same back-up planes are generated as the BPLANE contact; however, new contacts can be initialized during the simulation. This new initiation is especially important for air bags that have large internal open regions and where contacts between layers occur later during the

simulation time. Special care is taken into account to initialize the correct side of the primary contact face so that hooking behind layers is reduced to a minimum. Consequently, this type of contact is very CPU intensive. To this end, the new search algorithm can be skipped after a certain time has been reached. This time can be defined by the user with the TENDNEW entry on the CONTACT option. After this time has been reached, no new contact is searched for. Effectively, after the new TENDNEW is reached, the BPFULL contact algorithm behaves the same as the regular BPLANE contact algorithm.

6. An edge-to-edge contact is defined by specifying the surfaces of which you want the free edges to be included in the contact definition. The "free" edges are derived only from the supplied elements within those surfaces. This means that they are not necessarily the real free edges of the model.

   Limitation: when you want to use the edge-to-edge contact, you need to define a contact entry that has both the primary contact and the secondary contact surface in contact be of the type SURFEDGE. Of course, you can combine "normal" contact definitions with the edge-to-edge definition. Please note that each contact must be separately defined as a contact specification.

   You can influence the behavior of the searching algorithm that defines the edges that are in contact. By default, the edge contact is recognized when the direction of the velocity of the point that has potential contact is in the direction of the normal to the edge. This criterion may be too strict for some occasions. Therefore, with the EVIEW definition on the contact specification you may define an angle that defines a 3-D cone with the edge normal as the center line. When the velocity vector of a point searching for contact with an edge falls within this 3-D cone, the point is considered for contact with the edge. If the velocity vector of the point falls outside the 3-D cone, the point is ignored in the contact search. You can use this parameter when you encounter spurious edge-to-edge contact situations.

7. In case of V4 contact, REVERSE=ON takes effect only for edge-to-edge and primary-secondary contacts. For primary-secondary contact, it is only necessary to align the face normal when the SIDE entry is TOP or BOTTOM. For other situations, in principle, it is not necessary without violating the contact constraints.

8. TENDNEW is used only when the BPFULL contact search algorithm is used. After the TENDNEW time is reached, no new contacts are searched for.

9. A detailed description of the secondary contact node (de)activation methods is given here. These methods are only available for VERSION = V4:

   When a primary contact surface might fold onto itself, is more suited for eroding primary-secondary contact behaves as a single surface.

METHOD1	Nodes become active as secondary contact once they reside on the outside of the mesh. In case of primary-secondary contact, nodes on the primary contact surface will not act as a secondary contact. Nodes will be deactivated as secondary contact once all connected elements have failed.
METHOD2	Nodes become active as secondary contact once they reside on the outside of the mesh. In case of primary-secondary contact, nodes on the primary contact surface will not act as a secondary contact. Nodes will remain active as secondary contact once all connected elements have failed.
METHOD3	Nodes are active as secondary contact from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. In case of primary-secondary contact, nodes on the primary contact surface will not act as a secondary contact. Nodes will be deactivated as secondary contact once all connected elements have failed.
METHOD4	Nodes are active as secondary contact from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. In case of primary-secondary contact, nodes on the primary contact surface will not act as a secondary contact. Nodes will remain active as secondary contact once all connected elements have failed.

METHOD1A	For primary-secondary contact only. Nodes become active as secondary contact once they reside on the outside of the mesh. Nodes on the primary contact surface will also act as secondary contact, once they reside on the outside of the mesh. This method is more suited for eroding primary- secondary contact than METHOD1. Nodes will be deactivated as secondary contact once all connected elements have failed.
METHOD2A	For primary-secondary contact only. Nodes become active as secondary contact once they reside on the outside of the mesh. Nodes on the primary contact surface will also act as secondary contact, once they reside on the outside of the mesh. This method is more suited for eroding primary- secondary contact than METHOD2. Nodes will remain active as secondary contact once all connected elements have failed.
METHOD3A	For primary-secondary contact only. Nodes are active as secondary contact from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. Nodes on the primary contact surface will also act as secondary contact, once they reside on the outside of the mesh. This method is more suited for eroding primary-secondary contact than METHOD3. Nodes will be deactivated as secondary contact once all connected elements have failed.
METHOD4A	For primary-secondary contact only. Nodes are active as secondary contact from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. Nodes on the primary contact surface will also act as secondary contact, once they reside on the outside of the mesh. This method is more suited for eroding primary-secondary contact than METHOD4. Nodes will remain active as secondary contact once all connected elements have failed.

To choose the correct secondary contact activity switch, you can use the following flow schemes. The selection of a method depends on the desired results, and can be captured by three questions:

a. Only nodes on the outside of the mesh are active?

   In most cases only the secondary contact nodes on the outside need to be active.

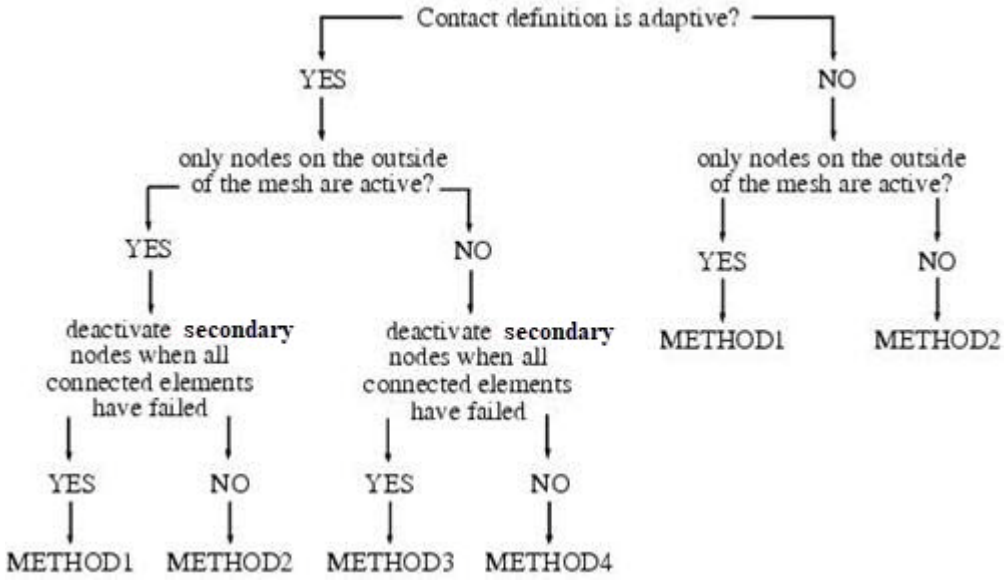
   In cases of high-velocity impact, it might be necessary to activate the internal secondary contact nodes also. This will prevent missing contacts for secondary contact nodes that move across the monitoring region of the primary contact face during the time-step it is activated.

b. Deactivate secondary contact nodes when all connected elements have failed?

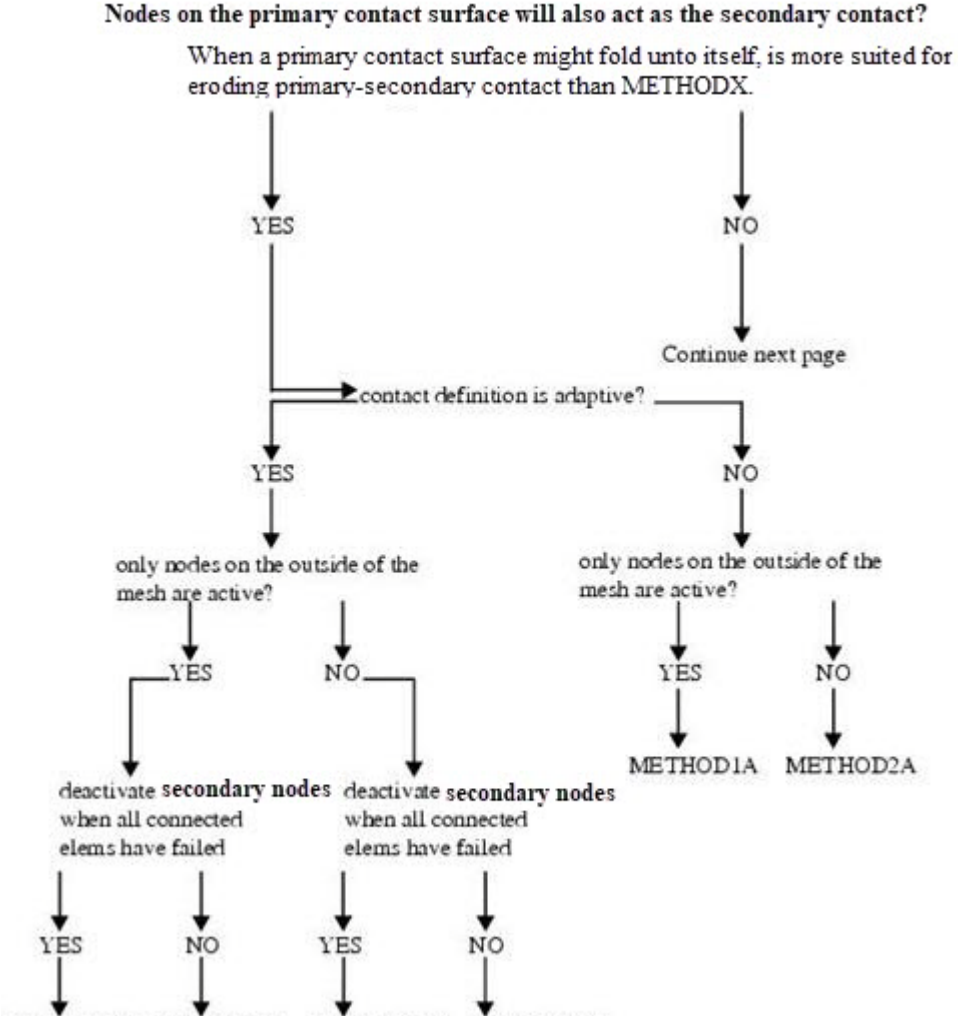
   This determines whether secondary contact nodes will remain active after all its connected elements have failed. This option only applies to an adaptive contact.

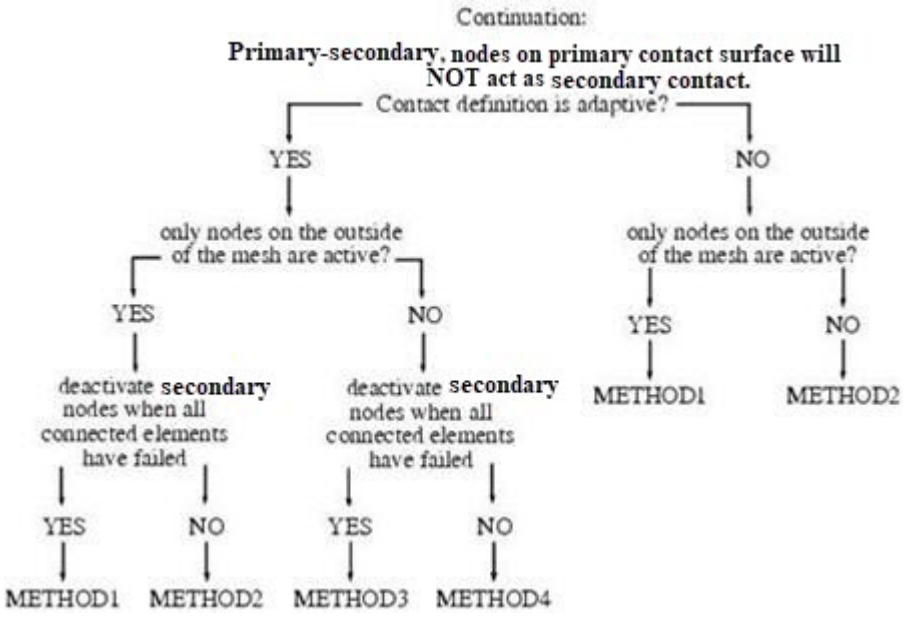
c. Nodes on the primary contact surface will also act as secondary contact? When a primary contact surface might fold onto itself this will prevent the primary contact surface from penetrating itself. Therefore the primary contact surface will behave as a single surface.

Flow scheme for a single surface contact:



Flow scheme for a primary-secondary contact:




Continuation:
Primary-secondary, nodes on primary contact surface will NOT act as secondary contact.
Contact definition is adaptive?
YES
NO
only nodes on the outside of the mesh are active?
YES
NO
deactivate secondary nodes when all connected elements have failed
YES
NO
METHOD1
METHOD2
only nodes on the outside of the mesh are active?
YES
NO
METHOD1
METHOD2
deactivate secondary nodes when all connected elements have failed
YES
NO
METHOD3
METHOD4

CONTFORC Contact Force Definition Using Force-Deflection Curves

The contact force is determined by force-deflection curves for loading and unloading. Damping can be specified either as a constant value or as a tabular function.

Format and Example

1	2	3	4	5	6	7	8	9	10
CONTFORC	CID	K	LOAD	UNLOAD	B-CONST	B-TABLE			
CONTFORC	9	1.E6				212			

Field	Contents	Type	Default
CID	Unique CONTFORC number, referenced from CONTACT entry.	I > 0	Required
K	Constant value for the contact stiffness.	R ≥ 0	See Remark 1.
	The contact force is calculated as:		
	$F_{contact} = Kd$		
	where d is the penetration depth. The force acts in the direction normal to the primary contact face. The same value is used during loading and unloading.		
LOAD	Number of a TABLED1 entry specifying the force versus penetration depth to be used when penetration increases. This is the loading phase.	I > 0	See Remark 1.
UNLOAD	Number of a TABLED1 entry specifying the force versus penetration depth to be used when penetration decreases. This is the unloading phase.	I > 0	Table number specified under LOAD
	By choosing a different unloading than loading curve, hysteresis can be modeled.		
B-CONST	Constant value of damper stiffness.	R ≥ 0	See Remark 2.
	The damper acts on the velocity difference between the face of the primary contact and the node of the secondary contact in the direction normal to the primary contact face.		
B-TABLE	Number of a TABLED1 entry specifying the damper stiffness.	I > 0	See Remark 2.
	The damper acts on the velocity difference between the face of the primary contact and the node of the secondary contact in the direction normal to the primary contact face.		

Remarks

1. Either K or LOAD must be specified.
2. None or just one of the options B-CONST, or B-TABLE must be specified.

CONTINI

Contact Initialization for In-Plane Folded Air Bags

Defines the initial contact state between two SUBSURF entries. Used for contact initialization of in-plane folded air bags.

Format and Example

1	2	3	4	5	6	7	8	9	10
CONTINI	CID	INIID	SUBID1	SUBID2		LEVEL	SIDE	REVERSE	
CONTINI	1	79	53	54		1.0	BOTH	NO	

Field	Contents		Type	Default
CID	Unique number of a CONTINI entry.		I > 0	Required
INIID	Number of a set of CONTINI entries. INIID must be referenced from a CONTACTentry.		I > 0	Required
SUBID1	Number of a secondary contact SUBSURF. The SUBSURF must be part of the secondary contact SURFACE, referenced on the CONTACT entry. (In case of a single surface contact, it must be part of that SURFACE.)		I > 0	Required
SUBID2	Number of a primary contact SURFACE. The SURFACE must be part of the primary contact SURFACE, referenced on the CONTACT entry. (In case of a single surface contact, it must be part of that SURFACE.		I > 0	Required
LEVEL	Defines the LEVEL of a contact initialization.		R ≥ 0.0	1.0 See Remark 2.
SIDE	Defines the side of the contact that will be accepted.		C	See Remark1.
	BOTH	Contact from both sides is accepted.		
	TOP	Only contact from the TOP side is accepted.		
	BOTTOM	Only contact from the BOTTOM side is accepted.		
REVERSE	Defines if the reverse CONTINI must be generated:		C	NO
	YES	A reversed CONTINI is generated with SUBID2 as the secondary contact and SUBID1 as the primary contact:		
		CONTINI,-,INIID,SUBID2,SUBID1		
	NO	A reversed CONTINI is not generated.		

Remarks

1. By default the SIDE is equal to the SIDE as specified on the CONTACT entry.
2. It is allowed to have multiple CONTINIs defined for a secondary contact subsurface. If a grid point of a secondary contact subsurface (SUBID1) finds a contact in more than one primary contact subsurface, only the ones with the highest level are accepted.

   For example, suppose a subsurf is used as secondary contact in three CONTINI definitions:

   ```
   CONTINI,1,101,SUBID1,SUBID2,,LEVEL1
   CONTINI,2,101,SUBID1,SUBID3,,LEVEL2
   CONTINI,3,101,SUBID1,SUBID4,,LEVEL3
   ```

   When a secondary contact node of SUBSURF,SUBID1 finds a contact in all three primary contact SUBSURFs. The following logic applies:

   ```
   LEVEL1=LEVEL2=LEVEL3 → all contacts are accepted
   LEVEL1>LEVEL2,LEVEL3 → only contact 1 is accepted
   LEVEL1=LEVEL2>LEVEL3 → only contact 1 and 2 are accepted
   LEVEL1,LEVEL2<LEVEL3 → only contact 3 and 2 are accepted
   LEVEL1<LEVEL2>LEVEL3 → only contact 2 is accepted
   ```
3. The options LEVEL, SIDE, and REVERSE are only valid for contact version V4.

CONTREL

Contact with Rigid Ellipsoids

Defines contact between rigid ellipsoids and Lagrangian grid points or rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
CONTREL	CID	SIDC	TYPE	SID	ARF				
CONTREL	20	30	GRID	40					

Field	Contents		Type	Default
CID	Unique contact number.		I > 0	Required
SIDC	Number of a SETC entry giving a list of the names of rigid ellipsoids on which contact can occur.		I > 0	Required
TYPE	The type of entity that can contact the rigid ellipsoids.		C	Required
	GRID	Grid points.		
	RIGID	Rigid bodies.		
SID	The number of a SET1 entry listing the grid points or rigid bodies that can contact the rigid ellipsoids. (See also Remark 2.)		I > 0	Required
ARF	Artificial restoration factor. This is the factor by which penetrated grid points are moved back to the surface of the ellipsoids. A value of 0 indicates that they are not moved. A value of 1 indicates that they are moved all the way back to the surface of the ellipsoid.		$0.0 \le R \le 1.0$	See Remark 1.

Remarks

1. For grid points attached to Lagrangian elements, the default for ARF is 1.0. For rigid surfaces, the default is 0.1.
2. Only rigid surfaces can be defined in the SET1 entry and are referenced by their number. MATRIG and RBE2-FULLRIGs cannot be referenced by this entry. Use the CONTACT entry instead.

CORD1C Cylindrical Coordinate System Definition, Form 1

Defines up to two cylindrical coordinate systems per entry by referencing three grid points that define a coordinate system. The grid points must be defined in a coordinate system other than the coordinate system that is being defined. The first grid point is the origin, the second lies on the z-axis, and the third lies in the plane of the azimuthal origin.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD1C	CID	G1	G2	G3	CID2	G4	G5	G6	
CORD1C	3	16	321	19					

Field	Contents	Type	Default
CID	Coordinate system number	I > 0	Required
G1, G2, G3	Grid point numbers G1, G2, and G3. The grid point numbers must be unique.	I > 0	Required
CID2	Optional second coordinate-system number.	I > 0	Blank
G4, G5, G6	Grid point numbers G4, G5, and G6. The grid point numbers must be unique.	I > 0	Blank

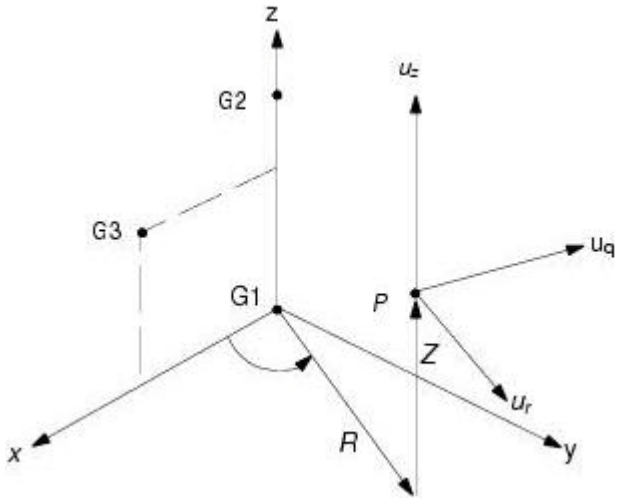


Figure 5-3 CORD1C Definition

Remarks

1. All coordinate system numbers must be unique.
2. The three grid points `G1`, `G2`, and `G3` must not be colinear.
3. The location of a grid point in the coordinate system is given by (R, θ, Z) where θ is measured in radians.
4. The velocity component directions at P depend on the location of P as shown above by U_r, U_θ, and U_z, when the coordinate system is used in a motion prescription.
5. One or two coordinate systems may be defined on a single line.

CORD1R Rectangular Coordinate System Definition, Form 1

Defines up to two rectangular coordinate systems per entry by referencing three grid points that define a coordinate system. The grid points must be defined in a coordinate system other than the coordinate system that is being defined. The first grid point is the origin, the second lies on the z-axis, and the third lies in the x-z plane.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD1R	CID	G1	G2	G3	CID2	G4	G5	G6	
CORD1R	3	16	321	19					

Field	Contents	Type	Default
CID	Coordinate system number.	I > 0	Required
G1, G2, G3	Grid point numbers G1, G2, and G3. The grid points must be unique.	I > 0	Required
CID2	Optional second coordinate system number	I > 0	Blank
G4, G5, G6	Grid point numbers G4, G5, and G6. The grid point numbers must be unique.	I > 0	Blank

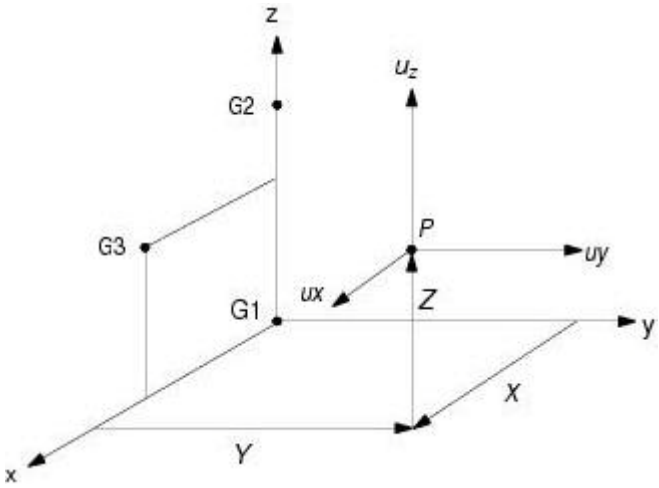


Figure 5-4 CORD1R Definition

Remarks

1. All coordinate system numbers must be unique.
2. The three grid points `G1`, `G2`, and `G3` must not be collinear.
3. The location of a grid point in this coordinate system is given by (X, Y, Z).
4. The velocity component directions at P depend on the location of P as shown above by U_x, U_y, and U_z, when the coordinate system is used in a motion prescription.
5. One or two coordinate systems may be defined on a single line.

CORD1S Spherical Coordinate System Definition, Form 1

Defines up to two spherical coordinate systems per entry by referencing three grid points that define a coordinate system. The grid points must be defined in an independent coordinate system. The first grid point is the origin. The second lies on the z-axis. The third lies in the plane of the azimuthal origin.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD1S	CID	G1	G2	G3	CID2	G4	G5	G6	
CORD1S	3	16	321	19					

Field	Contents	Type	Default
CID	Coordinate system number.	I > 0	Required
G1, G2, G3	Grid point numbers G1, G2, and G3. The grid points must be unique.	I > 0	Required
CID2	Optional second coordinate-system number.	I > 0	Blank
G4, G5, G6	Grid point numbers G4, G5, and G6. The grid points must be unique.	I > 0	Blank

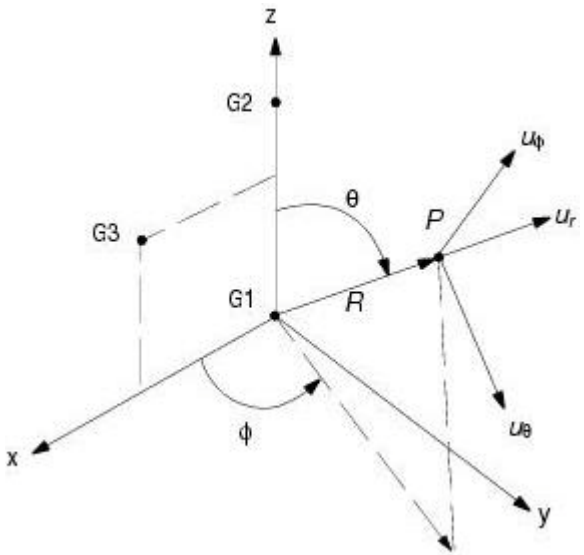


Figure 5-5 CORD1S Definition

Remarks

1. All coordinate system numbers must be unique.
2. The three grid points `G1`, `G2`, and `G3` must not be collinear.
3. The location of a grid point in this coordinate system is given by (R, θ, ϕ) where θ and ϕ are measured in degrees.
4. The velocity component directions at P depend on the location of P as shown above by U_r, U_θ, and U_ϕ, when the coordinate system is used in a motion prescription.
5. Grid points on the polar axis may not have their displacement directions defined in this coordinate system, since an ambiguity results.
6. One or two coordinate systems may be defined on a single line.

CORD2C Cylindrical Coordinate System Definition, Form 2

Defines a cylindrical coordinate system by referencing the coordinates of three grid points. The first point defines the origin, the second defines the direction of the z-axis, and the third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD2C	CID	RID	A1	A2	A3	B1	B2	B3	+
CORD2C	3	17	-2.9	1.0	0.0	3.6	0.0	1.0	+
+	C1	C2	C3						
+	5.2	1.0	-2.9						

Field	Contents	Type	Default
CID	Coordinate system number	I > 0	Required
RID	Reference coordinate system that is defined independent of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1, C2, C3	Coordinates of three points in the coordinate system referenced by RID.	R	0.0

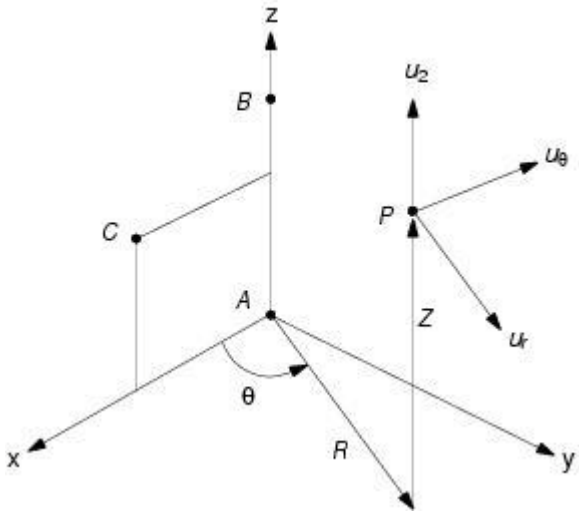


Figure 5-6 CORD2C Definition

Remarks

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate system numbers must be unique.
4. The location of a grid point in the coordinate system is given by (R, θ, Z) where θ is measured in degrees.
5. The velocity component directions at P depend on the location of P as shown above by U_r, U_θ, and U_z when the coordinate system is used in a motion prescription.
6. A RID of zero references the basic coordinate system.

CORD2R Rectangular Coordinate System Definition, Form 2

Defines a rectangular coordinate system by referencing the coordinates of three points. The first point defines the origin, the second defines the direction of the z-axis, and the third defines a vector that, with the z-axis, defines the x-z plane. The reference coordinate system must be independently defined.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD2R	CID	RID	A1	A2	A3	B1	B2	B3	+
CORD2R	3	17	-2.9	1.0	0.0	3.6	0.0	1.0	+
+	C1	C2	C3						
+	3.14	.1592	.653						

Field	Contents	Type	Default
CID	Coordinate system number.	I > 0	Required
RID	Reference coordinate system that is defined independent of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1, C2, C3	Coordinate of three points in the coordinate system referenced by RID.	R	0.0

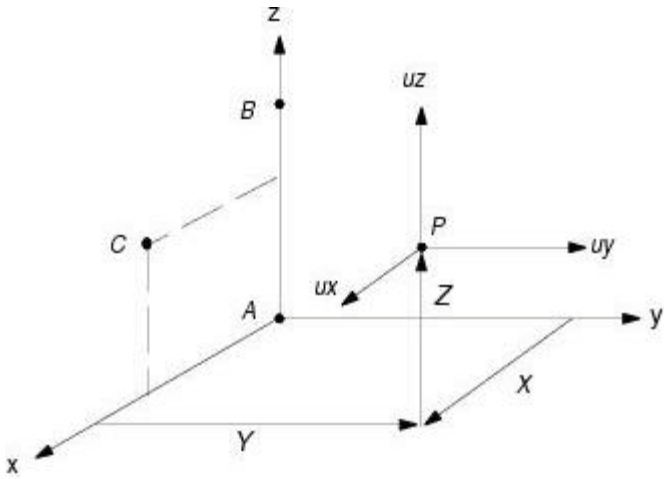


Figure 5-7 CORD2R Definition

Remarks

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate system numbers must be unique.
4. The location of a grid point in this coordinate system is given by (X, Y, Z).
5. The velocity-component directions at P depend on the location of P as shown above by U_x, U_y, and U_z, when the coordinate system is used in a motion prescription.
6. An RID of zero references the basic coordinate system.

CORD2S Spherical Coordinate System Definition, Form 2

Defines a spherical coordinate system by referencing the coordinates of three points. The first point defines the origin, the second defines the direction of the z-axis, and the third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD2S	CID	RID	A1	A2	A3	B1	B2	B3	+
CORD2S	3	17	-2.9	1.0	0.0	3.6	0.0	1.0	+
+	C1	C2	C3						
+	5.2	1.0	-2.9						

Field	Contents	Type	Default
CID	Coordinate system number.	I > 0	Required
RID	Reference coordinate system that is defined independently of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1, C2, C3	Coordinates of three points in the coordinate system referenced by RID.	R	0.0

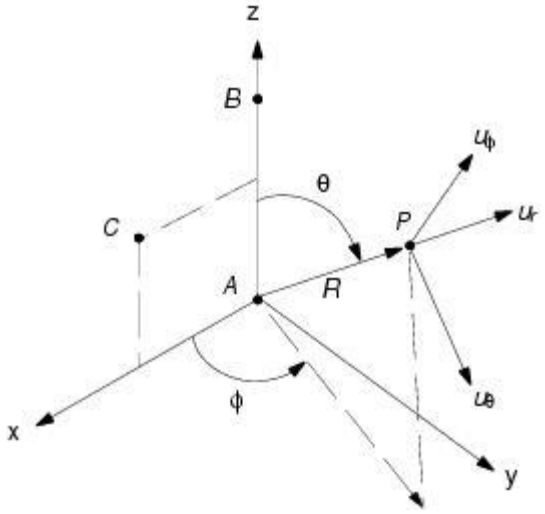


Figure 5-8 CORD2S Definition

Remarks

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3) and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate system numbers must be unique.
4. The location of a grid point in this coordinate system is given by (R, θ, ϕ) where θ and ϕ are measured in degrees.
5. The velocity components directions at P depend on the location of P as shown above by U_r, U_θ, and U_ϕ, when the coordinate system is used in a motion prescription.
6. Grid points on the polar axis may not have their displacement directions defined in this coordinate system, since an ambiguity results.
7. A RID of zero references the basic coordinate system.

CORD3R Moving Rectangular Coordinate System Definition, Form 1

Defines a rectangular coordinate system by referencing three grid points. The grid points must be defined in an independent coordinate system. The first grid point is the origin, the second lies on the z-axis, and the third lies in the x-z plane. The position and orientation of the coordinate system is updated as the grid points move.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD3R	CID	G1	G2	G3	CID	G1	G2	G3	
CORD3R	3	16	321	19					

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
G1, G2, G3	Grid-point numbers G1, G2, and G3 must be unique.	I > 0	Required

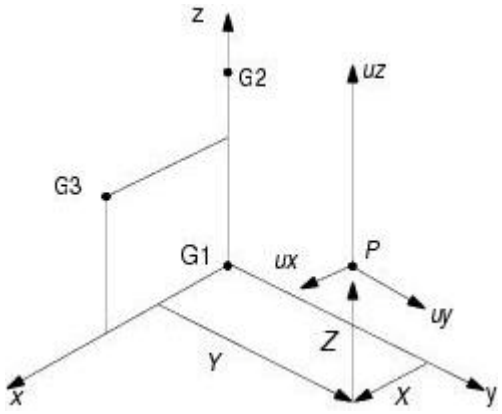


Figure 5-9 CORD3R Definition

Remarks

1. All coordinate system numbers must be unique.
2. The three points `G1`, `G2`, `G3` must not be collinear.
3. The location of a grid point P in this coordinate system is given by (X, Y, Z).
4. The displacement coordinate directions at P are shown above by ux, uy, and uz.
5. One or two coordinate systems may be defined on a single line.
6. The orientation of the coordinate system is updated each time step based on the motion of the grid points.

CORD4R Moving Rectangular Coordinate System Definition, Form 2

Defines a rectangular coordinate system by referencing the coordinates of three points. The first point defines the origin, the second defines the direction of the z-axis, and the third defines a vector that, with the z-axis, defines the x-z plane. The position and orientation of the coordinate system moves during the analysis by prescribed translation and rotation.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORD4R	CID	RID	A1	A2	A3	B1	B2	B3	+
CORD4R	3	17	-2.9	1.0	0.0	3.6	0.0	1.0	+
+	C1	C2	C3						+
+	5.2	1.0	-2.9						+
+	TTX	TTY	TTZ	TRX	TRY	TRZ			
+	33								

Field	Contents	Type	Default
CID	Coordinate system number.	I > 0	Required
RID	Reference coordinate system that is defined independently of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1,C2, C3	Coordinates of three points in the basic coordinate system.	R	0.0
TTX, TTY, TTZ	Number of TABLED1 entries defining the velocity of the origin of the coordinate system in the x-, y-, z-directions of the basic coordinate system.	I > 0	Fixed
TRX, TRY, TRZ	Number of TABLED1 entries defining the angular velocity of the coordinate system about the x-, y-, z-axes of the basic coordinate system.	I > 0	Fixed

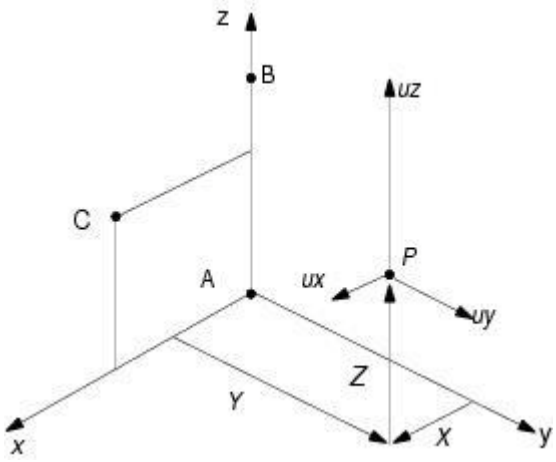


Figure 5-10 CORD4R Definition

Remarks

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate-system numbers must be unique.
4. The location of a grid point P in this coordinate system is given by (X, Y, Z).
5. The displacement coordinate directions at V are shown by ux, uy, and uz.

CORDROT

Corotational-frame Definition

Defines the direction of corotational axes in a material.

Format and Example

1	2	3	4	5	6	7	8	9	10
CORDROT	CID	G1	G2	G3					
CORDROT	100	1	2	3					

Field	Contents	Type	Default
CID	Unique coordinate-system number referred to by a DMAT or DMATEL Bulk Data entry.	I > 0	Required
G1, G2, G3	Relative grid-point numbers of elements of DMAT and DMATEL referring to this entry defining the orientation of the corotational frame. See Remark 5.	1 ≤ I ≤ 8	1,5,2

Remarks

1. The DMAT and DMATEL entries can refer to this type of coordinate system.
2. G1 defines the origin, G2 lies on the corotational z-axis, and G3 lies in the corotational (X-Z) plane.

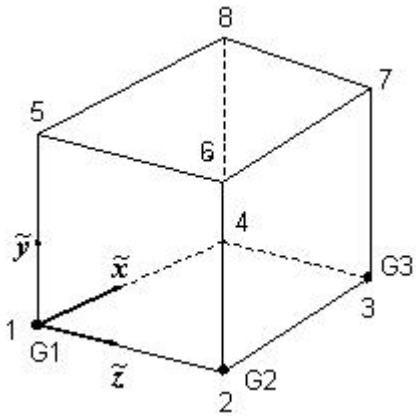


Figure 5-11 Element Corotational Frame According to the Example Given Above

3. The orientation of the element corotational frame is updated according to the spin of the element.
4. If the fields G1, G2, G3 are left blank, the default applies.
5. If it is referred by linear tetrahedron elements, the default of G1, G2, G3 is 1, 2, 4, respectively

COUCOHF Cohesive Friction Model to be Used for a COUPLE Entry

Defines a cohesive friction model suited for Euler Coupled analyses. The friction model is defined as part of the coupling surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
COUCOHF	CID	COHFRID	SUBID	COHFRICID					
COUCOHF	112	14	12	12					

Field	Contents	Type	Default
CID	Unique number of a COUCOHF entry.	I > 0.0	
COHFRID	Number of a set of COUCOHF entries. COUCOHF must be referenced from a COUPLE entry.	I > 0.0	
SUBID	Number of a SUBSURF, which must be a part of the SURFACE referred to from the COUPLE entry.	R ≥ 0.0	
COHFRICID	Cohesive friction ID	R ≥ 0.0	

Remarks

1. One couple entry can reference more than one COUCOHF entry. This allows a cohesive friction that varies along the coupling surface.
2. When SUBID is left blank, then the cohesive friction applies to the whole coupling surface.
3. A coupling surface segment can only have one porosity or one cohesive friction definition assigned.

COUHTR Heat Transfer Model to be Used with COUPLE Entry

Defines an heat transfer model suited for Euler Coupled analyses. The heat transfer model is defined as part of the coupling surface.

Format and Example

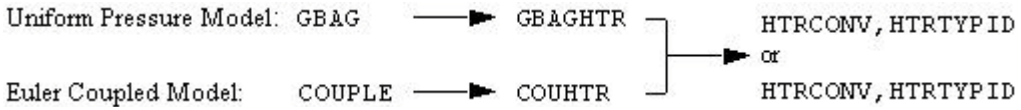
1	2	3	4	5	6	7	8	9	10
COUHTR	CID	HTRID	SUBID	HTRTYPE	HTRTYPID	COEFF	COEFFV		
COUHTR	100	1	2	3					

Field	Contents		Type	Default
CID	Unique number of a COUHTR entry		I > 0	Required
HTRID	Number of a set of COUHTR entries HTRID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the COUPLE entry.	I ≥ 0	0
	= 0	COUHTR definitions are used for the entire SURFACE referred to from the COUPLE entry.		
HTRTYPE	Defines the type of heat transfer		C	
	HTRCONV	The HTRCONV logic is used to model heat transfer through convection in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is trans ported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 exposes the complete subsurface area, while a value of COEFFV = 0.0 results in no heat transfer through the subsurface.		
	HTRRAD	The HTRRAD logic is used to model heat transfer through radiation in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is transported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 exposes the complete subsurface area, while a value of COEFFV = 0.0 results in no heat transfer through the subsurface.		

Field	Contents		Type	Default
HTRTYPID	ID of the entry selected under HTRTYPE, e.g. HTRCONV, HTRTYPID.		I	Required
COEFF	Method of defining the area coefficient.		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified on COEFFV.		
	TABLE	The area coefficient varies with time. COEFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEXentry depending on the COEFF entry.		0 < R < 1 or I > 0	1.0

Remarks

1. The same HTRTYPE entry referenced from this COUHTR entry can be referenced by a GBAGHTR entry. This allows for setting up the exact same model for either a uniform pressure model or an Euler Coupled model:



   This makes it possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).

2. A mixture of multiple COUHTR with different HTRTYPEs is allowed.
3. For the same SUBSURF, multiple different types of heat transfer may be defined.
4. A more detailed description can be found in *Dytran User's Guide*, Chapter 6: Air Bags and Occupant Safety, Porosity in Air Bags.

COUINFL Inflator Model to be Used with COUPLE Entry

Defines an inflator model suited for Euler Coupled analyses. The inflator model is defined as part of the coupling surface.

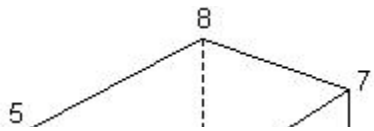
Format and Example

1	2	3	4	5	6	7	8	9	10
COUINFL	CID	INFID	SUBID	INFTYPE	INFTYPID	COEFF	COEFFV		
COUINFL	112	14	1204	INFLATR	80		0.012		

Field	Contents		Type	Default
CID	Unique number of a COUINFL entry.		I > 0	Required
INFID	Number of a set of COUINFL entries INFID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	Number of a SUBSURF, which must be a part of the SURFACE referred to from the COUPLEentry.		I > 0	Required
INFTYPE	Defines the type of inflator.		C	Required
	INFLATR	The INFLATR logic is used to model inflators in an air bag.		
	INFLATR1	The INFLATR1 logic is used to model inflators in an air bag.		
	INFLHY	The INFLHYB logic is used to model hybrid inflators in an air bag.		
	INFLHYB1	The INFLHYB1 logic is used to model hybrid inflators in an air bag.		
	INFLCG	The INFLCG logic models a cold gas inflator.		
INFTYPID	ID of the entry selected under INFTYPE; for example, INFLATR, INFTYPID.		I	Required
COEFF	Method of defining the area coefficient		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified on COEFFV.		
	TABLE	The area coefficient varies with time. COEFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0.0 < R < 1.0 or I > 0	1.0

Remarks

1. The INFLATR, INFLATR1, INFLHYB, or INFLHYB1 inflator geometry and location is defined by a subsurface (SUBID). The area of the hole through which the gas enters is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 opens up the complete subsurface area, while a value of COEFFV = 0.0 results in a closed inflator area with no inflow.
2. The same INFTYPE entry referenced from this COUINFL entry can be referenced by a GBAGINFL entry. This allows for setting up the exact same model for either a uniform pressure model or an Euler Coupled model:



   This makes it possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).
3. One couple entry can reference more than one COUINFL entry. This allows for modeling multiple inflators in an airbag module.

COUOPT

Coupling Options

Defines the interaction factor and a pressure load from the covered side acting on a SURFACE or SUBSURF.

Format and Example

1	2	3	4	5	6	7	8	9	10
COUOPT	CID	OPTID	SUBID	FACTOR	FACTORV				+
COUOPT	1	80	42	CONSTANT	0				+
+	PLCOVER	PLCOVERV							
+	CONSTANT	1.E5							

Field	Contents		Type	Default
CID	Unique number of a COUOPT entry.		I > 0	Required
OPTID	Number of a set of COUOPT entries. OPTID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be part of the SURFACE.	I ≥ 0	0
	= 0	COUOPT definitions used for the entire SURFACE.		
FACTOR	Method of defining the interaction FACTORV with which the Eulerian pressure acting on the SURFACE is multiplied.		C	CONSTANT
	CONSTANT	The FACTOR is constant and specified in FACTORV.		
FACTORV	The interaction factor		R	1
PLCOVER	Method of defining the pressure load exerted on the (SUB) SURFACE from the covered side. The pressure load is applied only when the Eulerian pressure is greater than zero.		C	CONSTANT
	CONSTANT	The PLCOVER is constant and specified in PLCOVERV.		
	TABLE	The PLCOVER varies with time. PLCOVERV is the number of aTABLED1 or TABLEEX entry giving the variation of the PLCOVER (y- value) with time (x-value).		
PLCOVERV	The pressure load or the number of a TABLED1 or TABLEEXentry depending on the PLCOVER entry.		R ≥ 0	0.

Remarks

1. The effect of specifying an interaction FACTOR is similar to specifying a porosity coefficient on a COUPOR entry. The difference is that in this case the SURFACE or SUBSURF still acts as a wall boundary for the Eulerian material.
2. PLCOVER works a follows. Consider, for example, a tank filled with high pressure gas. The pressure in the gas loads the tank and would cause the tank to expand very slightly. In addition, the tank is loaded by the ambient air pressure. This ambient air pressure counteracts the expansion due to the high pressure in the tank. COUOPT allows to take the effect of ambient air pressure into account. The PLCOVER pressure equals the ambient pressure.
3. The covered side of a SURFACE lies on the side where there is no Eulerian material. See *Dytran User's Guide*, Chapter 4: Fluid Structure Interaction, General Coupling.

COUP1FL

Coupling Surface Failure

Defines the surrounding variables when a segment of a coupling surface will fail.

Format and Example

1	2	3	4	5	6	7	8	9	10
COUP1FL	CFID	RHO	SIE	XVEL	YVEL	ZVEL	PRESSURE	MATERIAL	
COUP1FL	3	1.225	204082	900.					

Field	Contents	Type	Default
CFID	Unique number of a COUP1FL entry	I > 0	Required
RHO	Surrounding density	R > 0.0	See Remark 2.
SIE	Surrounding specific internal energy	R	See Remark 2.
XVEL	Surrounding x-velocity	R	See Remark 2.
YVEL	Surrounding y-velocity	R	See Remark 2.
ZVEL	Surrounding z-velocity	R	See Remark 2.
PRESSURE	Surrounding pressure	R	See Remark 4.

Remarks

1. This entry can only be used in combination with PARAM, FASTCOUP, FAIL and with either the HYDRO, MMHYDRO, or MMSTREN Euler Solver. For restrictions on the use of COUP1FL refer to param,flow-method. The coupling surface failure is associated with the element failure of the structure to which the surface is connected. Therefore, you have to define a failure model for the structure for the entry to take effect in the analysis.
2. For the first order Euler solvers, no restrictions apply to the values of the surrounding variables. For the Roe solver at least one of the surrounding variables should be defined. The default value of the density (RHO) will be set equal to the reference density as defined on the DMAT entry and the other variables (SIE, XVEL, YVEL, and ZVEL) are, by default, equal to zero.
3. The coupling surface must only consist of CQUADs and/or CTRIAs elements.
4. The PRESSURE field has to be left blank in combination with the Roe solver.
5. In combination with multi-material Euler, only outflow of material is allowed. Each material in an outflow Euler element is transported. The materials are transported in proportion to their relative volume fractions.

COUP1INT

Coupling Surface Interaction

Defines the interaction between two coupling surfaces.

Format and Example

1	2	3	4	5	6	7	8	9	10
COUP1INT	CIID	CID1	CID2						
COUP1INT	33	2	5						

Field	Contents	Type	Default
CIID	Unique number of a COUP1INT entry.	I > 0	Required
CID1	Number of COUPLE or COUPLE1 entry 1	I > 0	Required
CID2	Number of COUPLE or COUPLE1 entry 2	I > 0	Required

Remarks

1. This entry can only be used in combination with PARAM, FASTCOUP, , FAIL and with either the HYDRO, MMHYDRO or MMSTREN Euler Solver. For restrictions on the use of COUP1INT refer to PARAM, FLOW-METHOD. The interaction will be activated when failure of a Lagrangian structure on which the coupling surface lies is defined. Therefore, you have to define a failure model for the material of the structure.
2. The coupling surface should consist of CQUADs and/or CTRIAs.
3. For more detail on modeling flow between Eulerian domains, see PARAM, FLOW-METHOD.

COUPLE

General Euler-Lagrange Coupling Surface

Defines a coupling surface that acts as the interface between an Eulerian (finite volume) and a Lagrangian (finite element) domain.

Format and Example

1	2	3	4	5	6	7	8	9	10
COUPLE	CID	SID	COVER	REVERSE	CHECK	PORID	OPTID	CTYPE	+
COUPLE	100	37	INSIDE	ON	ON				+
+	INFID	HTRID	FS	FK	EXP	INTID			+
+			0.0	0.3	0.0				+
+	SET1ID	MESHID	TDEAC	COUP1FL	HYDSTAT	SKFRIC	COHFRID		
+									

Field	Contents		Type	Default
CID	Unique number of a COUPLE entry.		I > 0	Required.
SID	Number of a SURFACEentry defining the coupling surface.		I > 0	Required.
COVER	The processing strategy for Eulerian elements inside and outside of the coupling surface:		C	INSIDE
	INSIDE	The Eulerian elements inside the closed volume of the coupling surface are not processed.		
	OUTSIDE	The Eulerian elements outside the closed volume of the coupling surface are processed.		
	NONE	The Eulerian elements on both sides of the coupling surface are processed.		
REVERSE	Auto-reverse switch for the coupling surface segments:		C	ON
	ON	If required, the normals of the coupling surface segments are automatically reversed so that they all point in the same general direction as to give a positive closed volume.		
	OFF	The segments are not automatically reversed. The user is responsible for defining the correct general direction of the segment normals.		

Field	Contents		Type	Default
CHECK	Check switch for coupling surface segments:		C	ON
	ON	The normals of the segments are checked to verify that they all point in the same general direction and yield a positive closed volume.		
	OFF	The segments are not checked. It the responsibility of the user to ensure that the direction of the segment normals yield a positive closed volume.		
	When REVERSE is set to ON, the CHECK option is automatically activated.			
PORID	Number of a set of COUPOR entries that define the porosity for the SURFACE and/or SUBSURF entries.		I > 0	No porosity
OPTID	Not available for the Roe solver.		I > 0	No special options
	Number of a set of COUOPT entries that define special options for the SURFACE and the SUBSURF entries.			
CTYPE	Not available for the Roe solver. Coupling surface type definition:		C	STANDARD
	STANDARD	Standard Euler-Lagrange interaction.		
	AIRBAG	Coupling for air bag applications.		
	It is equivalent to the standard coupling algorithm with the following exceptions that tailor the solution for air bag applications: ■ Inflow through a porous (sub)surface occurs only when there is already some material (gas) in the Eulerian element. ■ Almost empty Eulerian elements will be automatically eliminated. The standard algorithm redistributes the small mass to the most suitable neighbor elements.			
INFID	Not available for the Roe solver. Number of a set of COUINFL entries that define the inflator(s) on the subsurface(s) of the coupling surface.		I > 0	No inflators
HTRID	Not available for the Roe solver. Number of a set of COUHTR entries that define the heat transfer definition(s) on the subsurface(s) of the coupling surface.		I > 0	No heat transfer
FS	Not available for the Roe solver. Static friction coefficient (See Remark 6.)		R ≥ 0	0.0

Field	Contents	Type	Default
FK	Not available for the Roe solver. Kinetic friction coefficient (See Remark 6.)	R ≥ 0	0.0
EXP	Not available for the Roe solver. Exponential decay coefficient (See Remark 6.)	R ≥ 0	0.0
INTID	ID of an INITGAS entry specifying initial gas composition for the Euler mesh (See Remark 7.)	I > 0	No initial gas composition
SET1ID	The number of a SET1 entry, which defined the Eulerian elements associated with this coupling surface.	I > 0	See Remark 8.
MESHID	The number of a MESH entry, which defines the Eulerian mesh associated with this coupling surface.	I > 0	See Remark 8.
TDEAC	Time of deactivation of the coupling surface and the associated Eulerian mesh.	R > 0.0	1.E20
COUP1FL	The number of a COUP1FL entry, which defines the surrounding variables for the coupling surface when its segments fail.	I > 0	See Remark 9.
HYDSTAT	The number of a HYDSTAT entry which specifies a hydrostatic preset. The preset is applied to all Euler elements specified by the SET1ID and MESHID.	I > 0	No hydrostatic preset
SKFRIC	Skin friction value.	R > 0.0	0.0 See Remark 11.
COHFRID	Number of a set of COUCOHF entries that define the cohesive friction on the subsurface(s) of the coupling surface.		

Remarks

1. All coupling surfaces must from a multi-faceted closed volume. If necessary, additional segments must be defined to achieve the closed volume. The closed volume must intersect at least one Euler element initially.
2. All segments must de attached to the face of an element. Dummy elements can be used to define any additional segments that are required to create the closed volume.
3. The normals of all segments that from the coupling surface must point in the same general direction and result in a positive closed volume. Setting the REVERSE option to ON ensures that this condition is satisfied, regardless of the initial definition of the segments.
4. The COVER field determines how Eulerian elements that are inside and outside of the coupling surface are processed. The default setting of INSIDE is appropriate for most of the problems. In the majority of analyses, the Eulerian material flows around the outside of the coupling surface. Therefore, the Eulerian elements that fall within the coupling surface do not contain material.

   For some specific applications, such as air bag inflation, the Eulerian material (gas) is completed contained within the coupling surface. In these cases, the COVER definition should be set to OUTSIDE.

5. When you want to use the fast coupling algorithm, you can define the parameter PARAM, FASTCOUP in the input file. The algorithm then used is substantially faster than the general coupling. The restriction is that you cannot use an arbitrarily shaped Euler mesh with the fast coupling algorithm. All element faces of the Euler mesh must have their normal pointing in any of the three basic coordinate directions. Thus, the mesh must be aligned with the three basic coordinate directions.
6. The friction model implemented for the coupling algorithm is a simple Coulomb friction definition. The friction coefficient μ is defined as:

   $$\mu = \mu_k + (\mu_s - \mu_k) \cdot e^{-\beta \cdot v}$$

   where μ_s is the static friction coefficient, μ_k is the kinetic friction coefficient, β the exponential decay coefficient and v the relative sliding at the point of contact.
7. An initial gas composition is for use with the single-material hydrodynamic Euler solver and an ideal-gas equation of state (EOSGAM) only.
8. Multiple coupling surfaces are available when you associate one Eulerian domain with a single coupling surface by either using the SET1ID or the MESHID option. Note that only one of the two options may be set and work only in combination with the fast coupling algorithm defined by PARAM,FASTCOUP.
9. The COUP1FL option is available and valid only in combination with the fast coupling algorithm with the failure option (PARAM,PARAM, ,FAIL). If no number is given the default values of the surrounding variables are used; the density (RHO) is set equal to the reference density as defined on the DMAT entry. By default, the other variables (SIE, XVEL, YVEL, and ZVEL are set equal to zero.
10. If an ACTIVE entry is present, its definition is ignored in case the TDEAC value is defined in combination with the fast coupling algorithm (PARAM,FASTCOUP).
11. The skin friction is defined as:

    $$C_f = \frac{\tau_w}{(0.5*\rho*u^2)}$$

    Here, τ_w denotes the shear friction in an Euler element adjacent to a couple surface segment where ρ is the density and u is the tangential relative velocity in the Euler element that is adjacent to a couple surface segment. SKFRIC is only used when VISC has been set on either an EOSGAM or an EOSPOL entry. If VISC has been set and if SKFRIC has not been set, a no slip condition will be prescribed at the interface between fluid and structure.
12. Option None requires the use of PARAM, AUTOCOUP.

COUPLE1

Euler-Lagrange Coupling Surface

Use COUPLEas this entry will be obsolete in the next release of Dytran.

Defines a coupling surface that acts as the interface between an Eulerian and a Lagrangian mesh for the Roe solver for single hydrodynamic materials.

Format and Example

1	2	3	4	5	6	7	8	9	10
COUPLE1	CID	SID	COVER	REVERSE	CHECK	PORID			+
COUPLE1	23	4							+
+	SET1ID	MESHID	TDEAC	COUP1FL					

Field	Contents		Type	Default
CID	Unique number of a COUPLEentry.		I > 0	Required
SID	Number of a SURFACE entry defining the coupling surface.		I > 0	Required
COVER	The processing strategy for Eulerian elements inside and outside of the coupling surface.		C	INSIDE
	INSIDE	The Eulerian elements inside the closed volume of the coupling surface are not processed.		
	OUTSIDE	The Eulerian elements outside the closed volume of the coupling surface are not processed.		
REVERSE	Auto reverse switch for coupling surface segments.		C	ON
	ON	If necessary, the normals of the coupling surface segments are automatically reversed so that they all point in the same general direction and give a positive closed volume.		
	OFF	The segment normals are not automatically reversed.		
CHECK	Checking switch for coupling-surface segments.		C	ON
	ON	The normals of the segments are checked to see whether they all point in the same general direction and give a positive closed volume.		
	OFF	The segment normals are not checked.		
	When REVERSE is set to ON, CHECK is automatically set to ON.			
PORID	Number of a set ofCOUPOR entries that define the porosity for the SURFACE and/or SUBSURF entries. Only the porosity models PORFLCPL or PORHOLE are supported.		I > 0	No porosity.

Field	Contents	Type	Default
SET1ID	The number of a SET1 entry, which defines the Eulerian region when multiple coupling surfaces are defined.	I > 0	See Remark 7.
MESHID	The number of a MESH entry, which defines the Eulerian region when multiple coupling surfaces are defined.	I > 0	See Remark 7.
TDEAC	Time of deactivation of the coupling surface and its Eulerian region.	R > 0.0	1.E20
COUP1FL	The number of a COUP1FL entry, which defines the surrounding variables for the coupling surface when its segments fail.	I > 0	See Remark 8.

Remarks

1. All coupling surfaces must form a multifaceted closed volume. If necessary, additional segments must be specified to achieve this.
2. All segments must be attached to the face of an element. Dummy elements must be used to define any additional segments that are required to create a closed volume.
3. The normals of all the segments that form the coupling surface must point in the same general direction and result in a positive closed volume. Setting the REVERSE field to ON ensures that this condition is satisfied, regardless of how the segments are defined initially.
4. The COVER field determines how Eulerian elements that are inside and outside of the coupling surface are processed. The default setting of INSIDE is appropriate for most problems, since in the majority of analyses, the Eulerian material flows around the outside of the coupling surface. Therefore, the Eulerian elements within the coupling surface are empty. However, for some specialized applications (such as air bag inflation), the Eulerian material is completely contained within the coupling surface, and in these cases COVER should be set to OUTSIDE.
5. For the fast coupling algorithm use PARAM,FASTCOUP.
6. The COUPLE1 entry can only be used in combination with PARAM,LIMITER,ROE.
7. Multiple coupling surfaces can be used defining one Eulerian region belonging to each coupling surface by setting the SET1ID or the MESHID option. Only one of the two options can be set and works only in combination with PARAM,FASTCOUP.
8. The COUP1FL option is only working in combination with PARAM,FASTCOUP, ,FAIL. If no number is given, the default values of the surrounding variables are given; the RHO is equal to the reference RHO on the DMAT entry and the other variables (SIE, XVEL, YVEL and ZVEL) are, by default, equal to zero.
9. The ACTIVE entry is ignored in case TDEAC is used in combination with PARAM,FASTCOUP.

COUPOR

Coupling Porosity

Defines porosity for SURFACE and SUBSURFentries used in general coupling.

Format and Example

1	2	3	4	5	6	7	8	9	10
COUPOR	CID	PORID	SUBID	PORTYPE	PORTYPID	COEFF	COEFFV		
COUPOR	111	203	31	PORFLOW	75		0.2		

Field	Contents		Type	Default
CID	Unique number of a COUPOR entry.		I > 0	Required
PORID	Number of a set of COUPOR entries. PORID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the COUPLE entry.	I ≥ 0	0
	= 0	COUPOR definitions used for the entire SURFACE referred to from the COUPLE entry.		
PORTYPE	Defines the type of porosity:		C	PORFLOW
	PORFLOW	The PORFLOW logic models a flow boundary in the coupling surface. The flow boundary acts on the open area of the coupling (sub)surface (SUBID). The open area is equal to the area of the (sub)surface multiplied by COEFFV. A hole can be modeled when COEFFV is set to 1.0. A closed area results for COEFFV = 0.0. The characteristics of the flow are defined on a PORFLOW entry, with ID as defined on the PORTYPID.		

Field	Contents		Type	Default
	PORHOLE	The PORHOLE logic models small holes in an air bag. A subsurface (SUBID) defines the hole. The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 opens up the complete area of the hole, while a value of COEFFV = 0.0 results in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORHOLE entry, with ID as defined on the PORTYPID.		
	PORLHOLE	The PORLHOLE logic can be used to model holes in an air bag. A subsurface (SUBID) defines the hole. The open area of the hole is equal to the area of the (sub) surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete area of the hole, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the velocity method. The characteristics for the flow are defined on a PORHOLE entry, with ID as defined on the PORTYPID.		
	PERMEAB	The PERMEAB logic models permeable air-bag material. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference between the gas inside the air bag and the environmental pressure. The function is specified on a PERMEAB entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		

Field	Contents		Type	Default
	PORFGBG	The PORFGBG logic models gas flow through a hole in the coupling surface connected to a GBAG. A subsurface (SUBID) defines the hole. The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 opens up the complete area of the hole, while a value of COEFFV = 0.0 results in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on aPORFGBG entry, with ID as defined on the PORTYPID.		
	PORFLGBG	The PORFLGBG logic can be used to model gas flow through a large hole in the coupling surface connected to a GBAG. A subsurface (SUBID) defines the hole. The open area of the hole is equal to the area of the (sub) surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete area of the hole, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the velocity method. The characteristics for the flow are defined on a PORFLGBG entry, with ID as defined on the PORTYPID.		
	PERMGBG	The PERMGBG logic models gas flow through a permeable area in the coupling surface connected to a GBAG. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference. This function is specified on a PERMGBG entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		

Field	Contents		Type	Default
	POREX	A EXPOR user subroutine defines a porosity model on the coupling surface. The volume and mass flow through the (sub)surface, the velocity of the flow, the pressure, the specific internal energy and the porosity coefficient COEFFV are all computed by the model programmed in the user subroutine. The name of the model to be used is defined on a POREX entry with the ID as specified on the PORTYPID field.	C	PORFLOW
	PORFCPL	The PORFCPL logic can be used to model gas flow through a hole in the coupling surface connected to another coupling surface. A subsurface (SUBID) defines the hole. The open area of the hole is equal to the area of the (sub) surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete area of the hole, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORFCPL entry, with ID as defined on the PORTYPID.		
	PORFLCPL	The PORFLGBG logic can be used to model gas flow through a large hole in the coupling surface connected to another coupling surface. A subsurface (SUBID) defines the hole. The open area of the hole is equal to the area of the (sub) surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete area of the hole, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the velocity method. The characteristics for the flow are defined on a PORFLCPL entry, with ID as defined on the PORTYPID.		
	PORHYDST	Prescribes a hydrostatic pressure profile		
PORTYPID	Number of a PORFLOW entry.		I > 0	Required

Field	Contents		Type	Default
COEFF	Method of defining the porosity coefficient.		C	CONSTANT
	CONSTANT	The porosity coefficient is constant and specified on COEFFV.		
	TABLE	The porosity coefficient varies with time. COEFFV is the number of a TABLED1 or TABLEEX entry defining the variation with time.		
COEFFV	The porosity coefficient or the number of a TABLED1or TABLEEX entry depending on the COEFF entry.		0 < R < 1. or I > 0	1.0

Remarks

1. A mixture of multiple COUPORs with different PORTYPEs is allowed.
2. All options of PORTYPE except PORFLOW, POREX, PORFCPL, and PORFLCPL can also be referenced by a GBAGPOR. This makes it possible to setup the exact same model for either a uniform pressure model (GBAG → GBAGPOR) or an Eulerian model (COUPLE→ COUPOR). It allows for setting up the model using a switch from full gas dynamics to uniform pressure (GBAGCOU).
3. The options PORFGBG and PERMGBG can be used to model air bags with different compartments.
4. The pressure, as defined by a PORFLOW or PORHOLE entry, is exerted on the Eulerian material. Similarly, the pressure in the connected GBAG, in case of a PORFGBG entry, is exerted on the Eulerian material. This pressure is applied over the open area only. The open area is equal to the area of the (sub)surface multiplied by COEFFV. The remaining closed area behaves as a wall boundary condition for the gas.
5. Any model that is not supported by the default types can be user-programmed in a subroutine called EXPOR.
6. To determine to use small hole definition (PORHOLE/PORFGBG/PORFCPL) or large hole definition PORLHOLE/PORFLGBG/PORFCPL) depends on the size of the hole relative to the size of the Euler elements. In general, when the size of the hole is 2-3 times larger than the Euler elements, use the velocity (or large hole) definition. When it is smaller use the pressure (or small hole) definition.

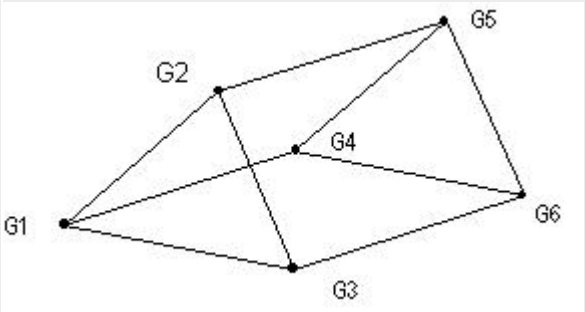
CPENTA

Solid Element with Six Grid Points

Defines a solid wedge element with six grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
CPENTA	EID	PID	G1	G2	G3	G4	G5	G6	
CPENTA	112	2	3	15	14	4	103	115	

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
PID	Number of a PSOLID or PEULER n property entry	I > 0	EID
G1–G6	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required
			

Remarks

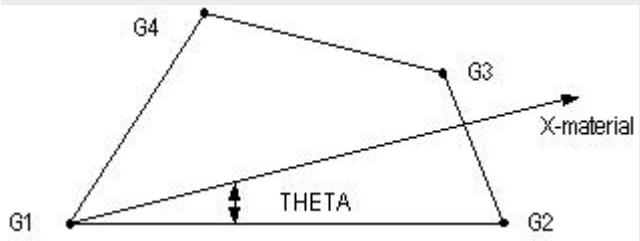
1. The element number must be unique with respect to all other elements.
2. G1, G2, and G3 must define a triangular face. G4, G5, and G6 define the opposite face with G1 opposite G4; G2 opposite G5, etc.
3. The faces may be numbered either clockwise or counterclockwise.
4. The Lagrangian CPENTA element performs poorly compared with the CHEXA element. This element should only be used where the geometry demands it, and it should be located well away from any critical areas. Always use the CHEXA element if you can.
5. The property number references a PSOLID or a PEULER entry. This determines whether the element is Lagrangian or Eulerian.

CQUAD4 Quadrilateral Element Connection

Defines a Lagrangian quadrilateral shell element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CQUAD4	EID	PID	G1	G2	G3	G4	THETA		+
CQUAD4	111	203	31	74	75	32			+
+			T1	T2	T3	T4			
+									

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
PID	Number of a PSHELL property entry	I > 0	EID
G1-G4	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required
THETA	Material property orientation specification (Real or blank; or 0 ≤ Integer < 1,000,000). If real, it specifies the material property orientation angle in degrees. If integer, the orientation of the material x-axis is along the projection onto the plane of the element of the x-axis of the coordinate system specified by the integer value. The figure below gives the sign convention for THETA.	I or R	
			
T1-T4	Thickness at the grid points G1 through G4.	R > 0.0	See Remark 4.

Remarks

1. The element number must be unique with respect to all other elements.
2. Grid points `G1` to `G4` must be ordered consecutively around the perimeter of the element.
3. If a CQUAD4 element has a thickness of 9999. (set on the PSHELL entry), it is not a shell element but it is converted to a CSEG entry. This conversion allows CSEGs to be easily defined using standard preprocessors. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Using a Modeling Program with Dytran for details.
4. If the four grid-point thicknesses are defined, the element thickness is the average of the defined thickness at the four grid points. If the thicknesses are not defined, the default thickness as specified on the PSHELLentry is used.
5. The `THETA` entry is only used with orthotropic materials.

CROD

Rod Element Connection

Defines a tension-compression element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CROD	EID	PID	G1	G2					
CROD	17	6	59	79					

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
PID	Number of a PROD, PBELT or PWELD property entry	I > 0	EID
G1, G2	Grid-point numbers of connected grid points	I > 0	Required
			

Remarks

1. Element numbers must be unique with respect to all other element numbers.
2. Only one rod element may be defined on a single line.

CSEG — Segment of a Contact Surface or Coupling Surface

Defines a segment with either three or four grid points.

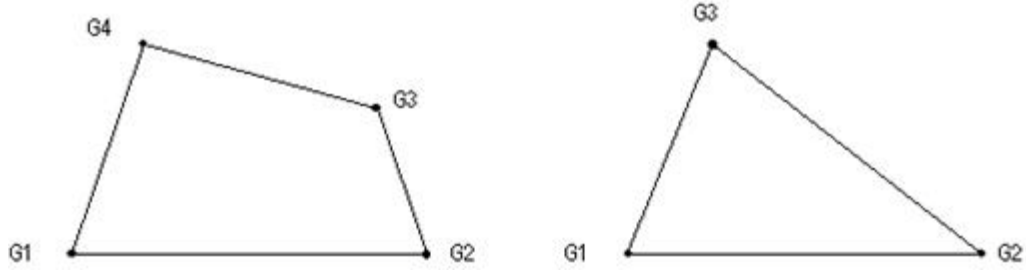
Format and Example

1	2	3	4	5	6	7	8	9	10
CSEG	ID	SID	G1	G2	G3	G4			
CSEG	101	17	13	19	64	63			

Field	Contents	Type	Default
ID	Unique segment number.	I > 0	Required
SID	Number of the set of segments to which this CSEG belongs.	I > 0	Required
G1-G4	Grid-point numbers defining the connectivity of the segment. For triangular segments, G4 should be blank or zero. All the grid points must be unique.	I > 0	Required

Remarks

1. The segment number must be unique with respect to all other segments.
2. Grid points G1 to G4 must be ordered consecutively around the perimeter of the element.



3. Segments can be automatically generated for shell and membrane elements, thereby saving the effort of creating several CSEG entries for contact surfaces and coupling with CQUAD4 and CTRIA3 elements. The elements for which segments are automatically generated are selected on SET1 entries referenced by the SURFACE entry.
4. To simplify the generation and checking of CSEG entries, CSEG entries can alternatively be defined using the CQUAD4 and CTRIA3 entries with a thickness of 9999. For details, see *Dytran User's Guide*, Chapter 9: Running the Analysis, Using a Modeling Program with Dytran.
5. Segments also can be defined using the CFACE and CFACE1 entries.

CSPR

Spring Connection

Defines a spring element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CSPR	EID	PID	G1	G2					
CSPR	2	6	9	33					

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
PID	Number of a PSPR property entry	I > 0	EID
G1, G2	Grid-point numbers at the ends of the spring. G1 must not be the same as G2	I > 0	Required

Remarks

1. The element number must be unique with respect to all other elements.
2. This entry defines a spring acting between two grid points. The force always acts in the direction of the line connecting the two grid points. The direction changes during the analysis as the grid points move.
3. The spring can have a linear or nonlinear force/deflection characteristic depending on the PSPR entry it references.

   Linear elastic with failure (PSPR)

   Nonlinear elastic (PSPR1)
4. CELAS1 and CELAS2define springs with a fixed orientation.

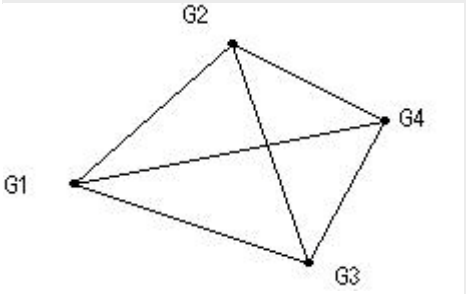
CTETRA

Solid Element with Four Grid Points

Defines a solid tetrahedral element with four grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
CTETRA	EID	PID	G1	G2	G3	G4			
CTETRA	112	2	3	15	14	4			

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
PID	Number of a PSOLID or PEULERn property entry	I > 0	EID
G1-G4	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required
			

Remarks

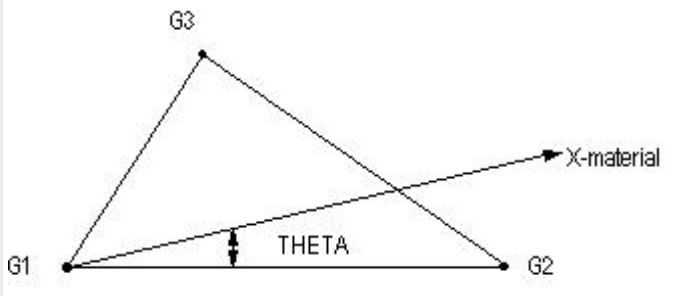
1. The element number must be unique with respect to all other element numbers.
2. The element can be numbered in any convenient order.
3. There are two types of Lagrangian CTETRA elements. The first one is based on linear tetrahedron FE formulation. The second one is based on collapsed CHEXA formulation that is the default due to upward compatibility reason. It performs poorly compared with the CHEXA element and should not be used unless absolutely necessary. It should be located well away from any area of interest.
4. The property number references a PSOLIDor PEULER entry. This determines whether the element is Lagrangian or Eulerian. To activate the linear tetrahedron FE formulation, IN and ISOP entry of the PSOLID must be set to 1.

CTRIA3 Triangular Element Connection

Defines a Lagrangian triangular shell or membrane element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CTRIA3	EID	PID	G1	G2	G3	THETA			+
CTRIA3	111	203	31	74	75				+
+			T1	T2	T3				
+									

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
PID	Number of a PSHELL property entry	I > 0	EID
G1-G3	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required
THETA	Material property orientation specification (Real or blank; or 0 ≤ Integer < 1,000,000). If real, specifies the material property orientation angle in degrees. If integer, the orientation of the material x-axis is along the projection onto the plane of the element of the x-axis of the coordinate system specified by the integer value. The figure below gives the sign convention for THETA.	I or R	
			
T1-T3	Thickness at the grid points G1 through G3.	R > 0.0	See Remark 3.

Remarks

1. The element number must be unique with respect to all other elements.
2. Grid points `G1` to `G3` must be ordered consecutively around the perimeter of the element.
3. If a `CTRIA3` element has a thickness of 9999 (set on the PSHELLn entry), it is not a shell element but is converted to a CSEG entry. This conversion allows CSEGs to be easily defined using standard preprocessors. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Using a Modeling Program with Dytran for details.

   If the three grid point thicknesses are defined, the element thickness is the average of the defined thickness at the three grid points.

   If the thicknesses are not defined, the default thickness as specified on the PSHELL entry is used.

CVISC

Damper Connection

Defines a viscous damper element.

Format and Example

1	2	3	4	5	6	7	8	9	10
CVISC	EID	PID	G1	G2					
CVISC	19	6	7	104					

Field	Contents	Type	Default
EID	Unique element number	I > 0	Required
PID	Number of a PVISC property entry	I > 0	EID
G1, G2	Grid-point numbers at the ends of the damper. G1 must not be the same as G2.	I > 0	Required

Remarks

1. The element number must be unique with respect to all other element numbers.
2. This entry defines a damper acting between two grid points. The force always acts in the direction of the line connecting the two grid points. The direction changes during the analysis as the grid points move.
3. The damper can have a linear or nonlinear force/velocity characteristic depending on the PVISC entry it references.

   Linear (PVISC)

   Nonlinear (PVISC1)
4. CDAMP1 and CDAMP2 define dampers with a fixed orientation.

CYLINDER

Defines the Shape of a Cylinder

Cylindrical shape used in the initial condition definition on the TICEUL entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
CYLINDER	VID		XC1	YC1	ZC1	XC2	YC2	ZC2	+
CYLINDER	4		0.	0.	0.	1.	1.	1.	+
+	RAD								
+	.5								

Field	Contents	Type	Default
VID	Unique cylinder number.	I > 0	Required
XC1, YC1, ZC1	Coordinates of point 1 (See Remark 1.)	R	Required
XC2, YC2, ZC2	Coordinates of point 2 (See Remark 1.)	R	Required
RAD	Radius of the cylinder.	R	Required

Remarks

1. A cylinder is defined by the two end points of the cylinder axis and a radius.
2. Initial conditions are defined for the elements that are fully or partially inside the cylinder. See *Dytran User's Guide*, Chapter 3: Constraints and Loading, Eulerian Loading and Constraints.
3. Also see theTICEUL Bulk Data entry.

DAREA

Concentrated Load or Enforced Motion

This entry is used in conjunction with a TLOAD entry and defines the location and direction of a concentrated load or enforced motion with a scale factor.

Format and Example

1	2	3	4	5	6	7	8	9	10
DAREA	LID	G	DIR	SCALE	G	DIR	SCALE		
DAREA	3	6	2	8.2	15	1			

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid point or rigid body where the load is applied	See Remark 5.	Required
DIR	Direction of the load. Enter 1, 2, or 3 to apply a loading in the x-, y-, or z-directions. Enter 4, 5, or 6 to apply loading about the x-, y-, or z-axes.	$1 \le I \le 6$	Required
SCALE	Scale factor for the load	R	1.0

Remarks

1. One or two loads can be defined on a line.
2. At time t, the load $F(t)$ is given by

   $$F(t) = \text{SCALE} * T(t)$$

   where SCALE is the scale factor and $T(t)$ is given by a table referenced from the TLOAD entry.
3. The load direction is defined in the basic coordinate system.
4. The direction of the load does not change during the analysis.
5. If G references a MATRIG, an RBE2 FULLRIG, or a RIGID surface, the load is applied to the center of the rigid body. If G references a MATRIG, G must be MR<id>, where id is the MATRIG number. If G references an RBE2 FULLRIG, G must be FR<id>, where id is the RBE2 number. If G references a RIGID surface, G is the RIGID surface number.
6. If the TYPE field on the TLOAD entry is 0, it defines a force or moment applied to a grid point. If the TYPE field is 2, it defines an enforced motion of the grid point. If the TYPE field is set to 12, it defines an enforced motion applied to the center of a rigid body. If the TYPE field is 13, it defines a force or moment applied to the center of a rigid body.

DETSPH

Spherical Detonation Wave

Defines the ignition point from which a spherical detonation wave travels, causing the reaction of high explosive materials.

Format and Example

1	2	3	4	5	6	7	8	9	10
DETSPH	DID	MID	X	Y	Z	VEL	TIME		
DETSPH	100	10	96.5	177.6	37.4	2379.	1.7E-6		

Field	Contents	Type	Default
DID	Unique detonation number	I > 0	Required
MID	Material number	I > 0	Required
X, Y, Z	Coordinates of the ignition point	R	0.0
VEL	Velocity of the detonation wave	R ≥ 0.0	0.0
TIME	Detonation time	R ≥ 0.0	0.0

Remarks

1. An element detonates when a spherical detonation wave originating from the detonation point at the specified time reaches the element.
2. By using a negative value for TIME, the detonation of EOSJWL material will be instantaneous and will become, in effect, an EOSGAM material. See the JWL Material Model vs. The Ideal Gas Approximation (EP3_7) example in Chapter 3: Fluid Dynamics of the *Dytran Example Problem Manual* for more details.

DMAT

General Constitutive Model

Defines a complete constitutive model as a combination of an equation of state, a shear model, a yield model, a failure model, a spall model (PMIN), and corotational frame.

Format and Example

1	2	3	4	5	6	7	8	9	10
DMA	MID	RHO	EID	SID	YID	FID	PID	CID	+
DMAT	22	3000.	100	109	307	308	402		+
+	BULKL	BULKQ			BULKTYP				
+									

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
EID	Number of an EOSxxx entry defining the pressure/density characteristic of the material		I > 0	Required
SID	Number of a SHRxxx entry defining the shear properties of the material		I ≥ 0	Hydrodynamic shear model
YID	Number of a YLDxxx entry defining the yield model for the material		I ≥ 0	Hydrodynamic yield model
FID	Number of a FAILxxx entry defining the failure model for the material		I ≥ 0	No failure
PID	Number of a PMINC entry defining the spallation characteristics of the material		I ≥ 0	See Remark 6.
CID	Number of a CORDROT entry. See Remark 7.		I ≥ 0	No corotational coordinate system
BULKL	Linear bulk-viscosity coefficient		R ≥ 0.0	0.0
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0.0	1.0
BULKTYP	Bulk viscosity type.		C	DYNA
	DYNA	Standard DYNA3D model		
	DYTRAN	Enhanced DYNA model		

Remarks

1. This material model can be used with Lagrangian and Eulerian solid elements and membrane elements.
2. If YID is blank or zero, a hydrodynamic yield model is used.
3. For Eulerian elements, if the TYPE field on the PEULER entry is set to HYDRO, then YID is either blank or references a YLDHY entry, and SID is blank. Conversely, if the TYPE field is set to STRENGTH, a non hydrodynamic yield model is specified.
4. This material is discussed in Materials.
5. Materials of shell elements need to be specified using the MAT1, MAT8, DMATEP, DYMAT24 or SHEETMAT entries.
6. If no PMINCentry is referenced, a minimum pressure of zero is assumed for the Eulerian elements, while spallation is prevented for the Lagrangian solid elements by assuming a minimum pressure of -1.E20. The PMINC entry will be ignored when a cavitation model through the EOSTAIT entry has been given.
7. The definition of a corotational coordinate system can only be used for Lagrangian solid elements. If no corotational coordinate system is specified, all stresses are in the basic coordinate system.
8. The failure model for Eulerian materials can be FAILEXor FAILMPS. For Lagrangian materials FAILMPS, FAILEX, FAILMES, and FAILSDTcan be addressed.
9. When PARAM,PMINFAIL is also set and a failure model is defined, Lagrangian solid elements also fail on the defined spallation pressure.

DMATEL

Isotropic Elastic Material Properties

Defines the properties of an isotropic elastic material for Lagrangian solid and membrane elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
DMATEL	MID	RHO	E	NU	G	K		CID	+
DMATEL	11	7850.0	210.E9	0.3					+
+	CSCALE	DMPFAC	BULKTYP	BULKQ	BULKL				
+									

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
E	Young's modulus.	R > 0.0	See Remark 1.
NU	Poisson's modulus.	R > 0.0	See Remark 1.
G	Shear modulus.	R > 0.0	See Remark 1.
K	Bulk modulus.	R > 0.0	See Remark 1.
CID	Number of a CORDROT entry. See Remark 5.	I > 0	No corotational coordinate system
CSCALE	When this material model is used with MEMB shell elements, the compressive stresses in the principal directions will be scaled with this factor.	R ≥ 0.0	1.0 See Remark 5.
	CSCALE=0.0 results in a tension only material.		
DMPFAC	When this material model is used with MEMB shell elements, damping is applied to the stresses:	R ≥ 0.0	0.0 See Remark 5.
	$d\sigma_{ij} = DMPFAC \cdot E \cdot \dot{\varepsilon}_{ij} \cdot dt_{elm}$		
	DMPFAC = 0.05 results in 5% damping applied to membrane stresses.		
BULKTYP	For Lagrangian solids only, Bulk-viscosity model:	C	DYNA
	DYNA Standard DYNA3D model.		
	DYTRAN Enhanced DYNA model.		
BULKQ	For Lagrangian solids only, Quadratic bulk-viscosity coefficient.	R ≥ 0.0	1.0

Field	Contents	Type	Default
BULKL	For Lagrangian solids only, Linear bulk-viscosity coefficient.	R ≥ 0.0	0.0
IIMMREL	Relaxation factor used with the Initial Metric Method. This option is only used when this material model is used with MEMB shell elements and the IMM method is activated.	0.0 < R < 1.0	1.0e-3. See Remark 7.

Remarks

1. Only two of the elastic constants E, Nu, G, and K should be defined.
2. The behavior of this material is discussed in Materials.
3. This material model can be used only with Lagrangian solid and membrane shell elements.
4. The definition of the corotational coordinate system can be used only for Lagrangian solid elements. If no corotational coordinate system is specified, all stresses are in the basic coordinate system.
5. For air bag modeling the following values of CSCALE and DMPFAC are suggested:

   ```
   CSCALE = 0.1
   DMPFAC= 0.05 to 0.20
   ```

6. The entry PMAXCUT, which was supported by Dytran V4.5 has become obsolete. A better methodology is now offered by scaling the compressive stresses in the principal directions, and using damping to suppress high-frequency oscillations (CSCALE and DMPFAC).
7. The Initial Metric Method relaxation factor is used to slow down the expansion of the membrane elements during the calculation. The default is sufficient in most simulations. The value of IMMREL is not used when the type of IMM is zero (see PARAM, IMM). The Initial Metric Method is described in the *Dytran User's Guide* in Initial Metric Method for Air Bags.

DMATEP

Elastoplastic Material Properties

Defines the properties of an isotropic-elastoplastic material for shell and beam elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
DMATEP	MID	RHO	E	ν	G	K	YID	FID	
DMATEP	11	7850.0	210.E9	0.3			100	101	

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
RHO	Density	R > 0.0	Required
E	Young's modulus	R > 0.0	See Remark 1.
ν	Poisson's ratio	0.5 ≥ R ≥ 0.0	See Remark 1.
G	Shear modulus	R ≥ 0.0	See Remark 1.
K	Bulk modulus	R ≥ 0.0	See Remark 1.
YID	Number of a YLDxxx entry defining the yield model for the material. (See Remark 6.)	I ≥ 0	See Remark 5.
FID	Number of a FAILxxx entry defining the failure model for the material. (See Remark 7.)	I	No Failure

Remarks

1. Only two of the elastic constants E, ν, G, or K should be defined.
2. The behavior of this material is discussed in Materials.
3. This material model can be used only with shell and beam elements.
4. If YID is 0 or blank, the material is elastic.
5. YID can refer to a YLDVM entry, in which case the material is elastoplastic with isotropic hardening, or for CQUADy and CTRIAz elements only, to a YLDJC entry to define a Johnson-Cook yield model.
6. If an elastoplastic material is specified for Belytschko-Schwer beams, a resultant plasticity model is used. The entire cross section yields at once.
7. The failure models that can be addressed by the DMATEP material definition are FAILMPS and FAILEX

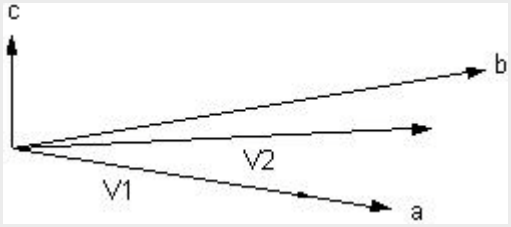
DMATOR

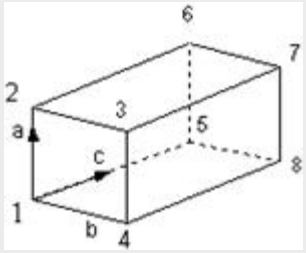
Orthotropic Elastic Material Properties

Defines an orthotropic elastic material for Lagrangian solid elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
DMATOR	MID	RHO	OPTION		FILE			FID	+
DMATOR	9	7800E-9	ELMAT		MAT.DAT			1	+
+	EA	EB	EC	NUBA	NUCA	NUCB			+
+	200E3	175.E3	105.E3	0.3	0.25	0.29			+
+	GAB	GBC	GCA						+
+	50E3	70E3	65.5E3						+
+	X1	Y1	Z1	X2	Y2	Z2			+
+									+
+			BULKTYP	BULKQ	BULKL				
+				1.2					

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
OPTION	Material axes option used to determine how the local material axis system is defined.		C	ELEM
	VECT	Globally orthotropic with the material axes defined by two vectors V1 and V2, specified using the fields X1, Y1, Z1 and X2, Y2, Z2. The a-axis is defined by the first vector. The b- and c-axes are then defined as:  		

Field	Contents		Type	Default
	ELEM	Globally orthotropic material with the material axes defined by element topology. The a, b, and c axes are defined as follows:  		
		Grid point 1 defines the origin, grid point 5 lies on the c-axis, and grid point 2 lies in the ac-plane.		
	ELMAT	Orthotropic material properties and the material coordinate system are defined by the element. The material properties are read from a file (formatted). The filename is specified in the sixth field of the first line.		
		Format of material properties file:		
		Record#		
		EID, DENSITY, DUMMY, DUMMY, DUMMY, $E_a, E_b, E_c, G_{ab}, G_{bc}, G_{ca},$ $\nu_{ab}, \nu_{ac}, \nu_{bc}, \nu_{ba}, \nu_{ca}, \nu_{cb}$		
	ELPROP	Globally orthotropic material with the material axes defined by element topology (see also ELEM). The elasticity matrix is available per element.		
FILE	Material file name (OPTION=ELMAT only).		C	
FID	Failure model number.		I > 0	No Failure
EA, EB, EC	Young's moduli in the a, b, and c directions.		R > 0.0	Required
NUBA, NUCA, NUCB	Poisson's ratio among the a, b, and c material directions. See Remark 6.		0.0 ≤ R ≤ 1.0	Required
GAB, GBC, GCA	Shear moduli among the three material directions.		R > 0.0	Required
X1, Y1, Z1	Components of the vector V1 in the basic coordinate system.		R	0.0
X2, Y2, Z2	Components of the vector V2 in the basic coordinate system.		R	0.0
BULKTYP	Bulk viscosity type:		C	DYNA

Field	Contents		Type	Default
	DYNA	Standard DYNA3-D model.		
	DYTRAN	Enhanced DYNA model.		
BULKL	Linear bulk-viscosity coefficient		R ≥ 0.0	0.0
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0.0	1.0

Remarks

1. The continuation line with bulk-viscosity data can be omitted.
2. The behavior of this material is discussed in Materials.
3. This material model can be used only with Lagrangian solid elements.
4. The failure models addressed by an orthotropic (DMATOR) material definition are FAILEX FAILEX1, FAILMES, FAILPRS, and FAILEST.
5. If FAILEX1, the extended user-defined failure, is used, set the OPTION to either ELMAT or ELPROP. The user-defined failure, FAILEX1, gives access to the material properties on an element basis.
6. The sum of any two ratios is less or equal to 1.

DRAWBD Drawbead definition

The grid points in drawbeads will be projected to MATRIG or a RIGID surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
DRAWBD	ID	SID	MID	DRWBDP	DRWBDRF	DRWBDUF	TID1	TID2	
DRAWBD	3	1	MR3	1.0	1.0E3	1.0E2			
+	NUMINT	BNDCNT							
+									

Field	Contents		Type
ID	Unique drawbead number. See Remark 1.	I > 0	Required
SID	SET1 ID for CROD that defines drawbeads. See Remark 2.	I > 0	Required
MID	Number, MR<number> which refers to MATRIG id or <number> referring to a RIGID surface id. See Remark 2.	I or C	Required
DRWBDP	Definition of drawbead depth. It increases the contact gap.	R ≥ 0.0	0.0
DRWBDRF	Drawbead restraining force per unit length. When DRWBEADF is defined in CONTACT, the values in DRAWBD will overwrite and DRWBEADF in CONTACT will be ignored.	R ≥ 0.0	Not used.
DRWBDUF	Drawbead uplift force per unit length.	R ≥ 0.0	0.0
TID1	TABLED1 ID for DRWBDRF. When TID1 is defined, DRWBDRF will be ignored. (Not available)	I > 0	Not used.
TID2	TABLED1 ID for DRWBDUF. When TID1 is defined, DRWBDUF will be ignored. (Not available)	I > 0	Not used.
NUMINT	The number of equally spaced integration points along the draw bead. (Not available)	I ≥ 0	1.0 (not used)
BNDCNT	Binding contact id. See Remark 3.	I ≥ 0	Not used

Remarks

1. DRAWBD will be used in the secondary contact definition only when VERSION is set to DRAWBDV4 in CONTACT definition.
2. The grid points of CROD's will be attached to MATRIG or a RIGID surface. When there is a gap between the grid point of CROD and the nearest element of MATRIG or the RIGID surface, the gap will be automatically removed.
3. When drawbead is located too close to the border of a blank holder, uplift forces are applied on grid points that are not contacted by both a die and a blank holder. This event may generate strange behavior. To prevent the problem, the opposite side of contact is used to apply uplift forces. When there is no opposite side contact, the uplift force is not applied. If the drawbead is located on a blank holder, the BNDCNT id must be the contact id between a blank and a die. If the drawbead is located on a die, the BNDCNT id must be the contact id between a blank and a blank holder.
4. CROD elements must have a different material than MID in the DRAWBD definition. The grid points of CROD elements will be projected and included into MID automatically.

DRAWBD1 Drawbead definition

Defines the drawbeads. The grid points in drawbeads will be projected to MATRIG or a RIGID surface. This can be used to merge multiple drawbeads into one contact.

Format and Example

1	2	3	4	5	6	7	8	9	10
DRAWBD1	ID	CSID	SID	MID	DRWBDP	DRWBDRF	DRWBDUF		
DRAWBD1	3	1999	1	MR3	1.0	1.0E3	1.0E2		
+	TID1	TID2	NUMINT	BNDCNT					
+									

Field	Contents	Type	Default
ID	Unique drawbead number. See Remark 1.	I > 0	Required
CSID	Secondary ID in CONTACT. See Remark 5.	I > 0	Required
SID	SET1 ID for CRODs that defines drawbeads. See Remark 2.	I > 0	Required
MID	Number, MR<number> which refers to MATRIG id or <number> referring to a RIGID surface id. See Remark 2.	I or C	Required
DRWBDP	Definition of drawbead depth. It increaseS the contact gap.	R ≥ 0.0	0.0
DRWBDRF	Drawbead restraining force per unit length. When DRWBEADF is defined in CONTACT, the values in DRAWBD1 will overwrite and DRWBEADF in CONTACT will be ignored.	R ≥ 0.0	Not used.
DRWBDUF	Drawbead uplift force per unit length.	R ≥ 0.0	0.0
TID1	TABLED1 ID for DRWBDRF. When TID1 is defined, DRWBDRF will be ignored. (Not available)	I > 0	Not used.
TID2	TABLED1 ID for DRWBDUF. When TID1 is defined, DRWBDUF will be ignored. (Not available)	I > 0	Not used.
NUMINT	Number of equally spaced integration points along the draw bead. (Not available)	I ≥ 0	1.0 (not used)
BNDCNT	Binding contact id. See Remark 3.	I ≥ 0	Not used

Remarks

1. DRAWDB1 will be used in the slave definition only when VERSION is set to DRAWBDV4 in the CONTACT definition.

2. The grid points of CRODs will be attached to MATRIG or a RIGID surface. When there is a gap between the grid point of CROD and the nearest element of MATRIG or RIGID surface, the gap will be automatically removed.
3. When drawbead is located to close to the border of a blank holder, uplift forces are applied on grid points that are not contacted by both of a die and a blank holder. This event may generate strange behavior. To prevent the problem, the opposite side of contact is used to apply uplift forces. When there is no opposite side contact, the uplift force is not applied. If the drawbead is located on a blank holder, BNDCNT id must be the contact id between a blank and a die. If the drawbead is located on a die, BNDCNT id must be the contact id between a blank and a blank holder.
4. CROD elements must have a different material than MID in DRAWBD1 definition. The grid points of CROD elements will be projected and included into MID automatically.
5. CSID can be used in multiple DRAWBD1 definitions and multiple DRAWBD1 can be used into one contact.

DYMAT14 Soil and Crushable Foam Material Properties

Defines a nonlinear material for Lagrangian solid elements that crushes under hydrostatic loading and is elastic-plastic under deviatoric loading. Material failure can be included.

Format and Example

1	2	3	4	5	6	7	8	9	10
DYMAT14	MID	RHO	G	K	TABLE	TYPE	VALUE	CUTOFF	+
DYMAT14	3	0.01	5.	3.	111	CRUSH	-100.	PFRAC	+
+	A_0	A_1	A_2	YIELD		YSTYP			+
+	1.	0.	0.	YSURF		DYNA			+
+			BULKTYP	BULKQ	BULKL				
+			DYNA	1.4	0.05				

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
G	Shear modulus		R > 0.0	Required
K	Bulk modulus		R > 0.0	Required
TABLE	Number of a TABLED1 entry giving the variation of pressure (y-value) with crush factor or volumetric strain (x-value).		I > 0	Required
TYPE	The type of data defined as the x value in the table:		C	CRUSH
	CRUSH	Crush factor (1 = relative volume)		
	STRAIN	Volumetric (true) strain		
	See Remark 3.			
VALUE	The value for the cut-off pressure		R < 0.0	See Remark 4.
CUTOFF	Cut-off pressure		C	PFRAC
	PFAIL	Pressure for total tensile failure		
	PFRAC	Pressure for tensile failure		
	PMIN	Minimum pressure		
A_0, A_1, A_2	Yield function constants.		R	0.0
YIELD	Surface description:		C	YSURF

Field	Contents		Type	Default
	YSURF	The yield surface (see Remark 7.) is defined as a function of p and J_2.		
	YSTRESS	The yield surface is defined as a function of p and s_y.		
YSTYP	Type of YSURF Yield Surface description:		C	DYNA
	DYNA	DYNA definition		
	DYTRAN	Dytran additional definition (See Remark 7.)		
BULKTYP	Bulk-viscosity model:		C	DYNA
	DYNA	Standard DYNA3D model		
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient		R ≥ 0.0	0.0

Remarks:

1. If BULKTYP, BULKQ, or BULKL are blank or zero the default values are used.
2. The continuation line with the bulk-viscosity data can be omitted.
3. The pressure-volume characteristic can either be defined in terms of the amount of crush, which is minus the engineering strain and is defined as (1 – V/V0), with V/V0 as the relative volume; or in terms of the volumetric (true) strain which is defined as:

   $$\int_{t_0} \frac{dV}{V} \text{ or, } ln(V/V_0).$$

   The crush must be between 0 and 1.

   The volumetric strain must always be negative.
4. If the field for the value of PFAIL/PFRAC/PMIN is left blank, then this value is calculated from the yield function defined by the constants A_0, A_1, and A_2. In case of a Mohr-Coulomb yield model, the cut-off pressure is calculated as the root of the pressure-yield stress curve. If the YSURF option is used, the cut-off pressure is calculated as the intersection point of the yield surface with the hydrostat (if only A_0 is nonzero, then the cut-off pressure is set to -100K, where K is the bulk modulus). The cut-off pressure must be negative.
5. Either a minimum pressure (PMIN) or a failure pressure (PFAIL or PFRAC) can be specified. In the first case, this corresponds to a tensile cutoff, where the pressure cannot fall below the minimum value. In the second case, if the pressure falls below the failure pressure the element fails and cannot carry tensile loading for the remainder of the analysis. Thus, the pressure can never become negative again. If PFAIL is used, the elements can physically fail, which means that the stresses are set to zero, but also the failure flag is used as in normal FAILxxx models. If PFRAC is used, only the stresses are set to zero.
6. This material can only be used with Lagrangian solid elements.

7. If the YSTRESS option is used, the yield stress is determined by a Mohr-Coulomb model:

$$\sigma_y = MIN(A_0 + A_1 p, A_2)$$

If the YSURF option is used, the yield surface in three-dimensional space is defined by $\Phi_s = 0$ here

$$\Phi_s = \frac{1}{2} s_{ij} s_{ij} - (B_0 + B_1 p + B_2 p^2) = J_2 - (B_0 + B_1 p + B_2 p^2)$$

where s_{ij} are the deviatoric stresses and J_2 is the second invariant of the stress deviation. The coefficients B_0, B_1, and B_2 can be related to the coefficients A_0, A_1, and A_2, which are defined on the DYMAT14 entry. The relation between the coefficients depends on the YSTYP field as shown below. If the YSTYP field is DYTRAN, then

$$B_0 = A_0$$

$$B_1 = A_1$$

$$B_2 = A_2$$

Thus, the yield stress (See: Constraints and Loading (Ch. 3) in the *Dytran User's Guide*) is defined as

$$\sigma_y = \sqrt{3(A_0 + A_1 p + A_2 p^2)}$$

If the YSTYP field is DYNA, then

$$B_0 = \frac{1}{3} A_0^2$$

$$B_1 = \frac{2}{3} A_0 A_1$$

$$B_2 = \frac{1}{3} A_1^2$$

and A_2 is ignored.

Thus, the yield stress is defined as

$$\sigma_y = A_0 + A_1 p$$

8. The behavior of this material is described in Materials (Ch. 3) in the *Dytran Theory Manual*.

DYMAT24 Piecewise Linear Plasticity Material

Defines the properties of a nonlinear, plastic material with isotropic hardening where the stress-strain curve is piecewise linear for shell, beam and Lagrangian solid elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
DYMAT24	MID	RHO	E	NU	TABLE	TYPE	TABY		+
DYMAT24	17	7850.	210.E9	0.3	39	ENG			+
+	YIELD	EH	EPSF	D	P	VOLF	EPSF-C		+
+			0.37	40.5	5	1.E-6			+
+			BULKTYP	BULKQ	BULKL				
+			DYNA	1.4	0.05				

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
E	Young's modulus		R > 0.0	Required
NU	Poisson's ratio		0.0 < R ≤ 0.5	Required
TABLE	Number of a TABLED1 entry giving the variation of effective stress (y-value) with effective strain (x- value).		I > 0	See Remark 3.
TYPE	The type of stress and strain defined in TABLE.		C	TRUE
	ENG	Engineering stress and strain.		
	TRUE	True stress and strain.		
	PLAST	True stress and plastic strain.		
	PMOD	Plastic modulus and plastic strain.		
TABY	Number of a TABLED1 entry giving the variation of the scale factor for the yield stress (y-value) with the strain rate (x-value). Strain rate effects can also be specified using the Cowper-Symonds relation. (See input parameters D and P.)		I > 0	See Remark 5.
YIELD	Yield stress		R > 0.0	See Remark 5.
EH	Hardening modulus		R > 0.0	See Remark 5.
EPSF	Plastic strain at failure		R > 0.0	No failure

Field	Contents	Type	Default
D	Factor D in the Cowper-Symonds rate enhancement equation	R ≥ 0.0	See Remark 5.
P	Factor P in the Cowper-Symonds rate enhancement equation	R ≥ 0.0	See Remark 5.
EPSF-C	Plastic strain at failure for material under compression	R > 0.0	EPSF
VOLF	If the volume of Lagrangian solid elements becomes less than VOLF, the element fails.	R ≥ 0.0	1.E-12
BULKTYP	Bulk viscosity model	C	DYNA
	DYNA Standard DYNA3D model		
BULKQ	Quadratic bulk-viscosity coefficient	R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient	R ≥ 0.0	0.0

Remarks:

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values apply.
2. The continuation line with the bulk-viscosity data can be omitted.
3. If TABLE is blank or zero, a bilinear stress-strain curve is assumed. If TABLE has a value, it refers to a TABLED1 entry giving the stress-strain curve for the material.
4. If TABLE is defined, the value of YIELD should be left blank, since it is determined from the stress-strain curve.
5. If TABY is blank or zero, and D and P are blank or zero, the yield stress does not vary with strain rate. If TABY has a value, then it references a TABLED1 entry, which gives the variation with strain rate of the scale factor applied to the yield stress (D and P must be blank or zero).

   If TABY is blank or zero and D and P are defined, the enhancement of the yield stress with strain rate is calculated as

   $$\frac{\sigma_d}{\sigma_y} = 1 + \left(\frac{\dot{\varepsilon}}{D}\right)^{1/P}$$

   where σ_d is the dynamic stress and σ_y is the static yield stress (YIELD), and $\dot{\varepsilon}$ is the equivalent plastic strain rate.
6. If TYPE is set to ENG or TRUE, Young's modulus is calculated from the stress-strain curve. When Young's modulus is specified together with TYPE set to ENG or TRUE, the calculated Young's modulus is substituted by the value specified.
7. The behavior of this material is described in Section Materials.
8. This material can only be used with Lagrangian solid, shell and beam elements.

DYMAT25

Cap Material Model

Defines a kinematic hardening Cap material model. The material can be used to model soil with compaction to capture a soft soil response. It can also be used to model materials like concrete or rock.

Format and Example

1	2	3	4	5	6	7	8	9	10
DYMAT25	MID	RHO	G	K	ALPHA	THETA	GAMMA	BETA	+
DYMAT25	101	2700.0	1.1E+10	1.4E+10	2.7E+7	0.11	8.0E+6	1.4E-7	+
+	R	D	W	X0	CBAR	N	TYPE	ITER	+
+	4.43	4.6E-10	0.42	1.1E+8			SOIL		+
+	TOFF								+
+	-1.0E+11								+
+			BULKTYP	BULKQ	BULKL				
+			DYNA	1.44	0.06				

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required.
RHO	Material density		R ≥ 0.0	Required.
G	Shear modulus		R ≥ 0.0	Required.
K	Bulk modulus		R ≥ 0.0	Required.
ALPHA	Failure envelope parameters		R ≥ 0.0	Required.
THETA	Failure envelope linear coefficient		R ≥ 0.0	Required.
GAMMA	Failure envelope exponential coefficient		R ≥ 0.0	See Remark 3.
BETA	Failure envelope exponent		R ≥ 0.0	Required.
R	Cap surface axis ratio		R ≥ 0.0	Required.
D	Hardening law exponent		R ≥ 0.0	Required.
W	Hardening law coefficient		R ≥ 0.0	Required.
X0	Hardening law exponent		R ≥ 0.0	0.0
CBAR	Kinematic hardening coefficient		R ≥ 0.0	0.0
N	Kinematic hardening parameter		R ≥ 0.0	0.0
TYPE	Formulation type:		C	SOIL
	SOIL	Soil or concrete (cap surface may contract)		
	ROCK	Rock (cap surface does not contract)		

Field	Contents		Type	Default
ITER	Iteration scheme:		C	VEC
	VEC	Fixed number of iterations (vectorized)		
	FULL	Fully iterative		
TOFF	Tension cut off (positive in compression)		R < 0.0	See Remark 5.
BULKTYP	Bulk viscosity model:		C	DYNA
	DYNA	Standard DYNA3D model		
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient		R ≥ 0.0	0.0

Remarks

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values apply.
2. The continuation lines with the bulk viscosity data may be omitted.
3. For a physically meaningful model, the value of the failure envelope exponential coefficient should be less than the failure envelope parameter $(\alpha < \gamma)$.
4. This material can only be used with Lagrangian solid elements.
5. The tension cut off value (T_{off}) can be defined on the entry and must be less than zero. If the tension cut off is left blank, Dytran calculates the tension cut off as the intersection point of the failure envelope surface with the J1-axis as described in the Theory Manual.
6. The behavior of this material is discussed in detail in the Theory Manual.

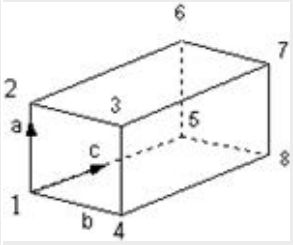
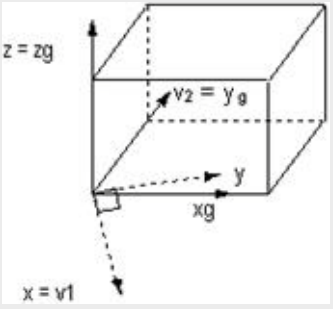
DYMAT26 Orthotropic Crushable Material Model

Defines the properties of an orthotropic, crushable material model for Lagrangian solid elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
DYMAT26	MID	RHO	E	NU	YIELD	RELV	TYPE	OPTION	+
DYMAT26	5	1800.	180.E9	0.3	180.E6	0.1	CRUSH	VECT	+
+	TIDXX	TIDYY	TIDZZ	TIDXY	TIDYZ	TIDZX	TIDSR		+
+	10	11	12	13	14	15	16		+
+	EXX	EYY	EZZ	GXY	GYZ	GZX			+
+	60.E9	70.E9	60.E9	20.E9	10.E9	15.E9			+
+			BULKTYP	BULKQ	BULKL				+
+			DYNA	1.4	0.05				+
+	NUYX	NUZX	NUZY						+
+	0.0	0.0	0.0						+
+	X1	Y1	Z1	X2	Y2	Z2			
+	0.	1.	1.	1.	1.	0.			

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
RHO	Density	R > 0.0	Required
E	Young's modulus for the fully compacted material	R > 0.0	Required
NU	Poisson's ratio for the fully compacted material	-1.0 < R < 0.5	Required
YIELD	Yield strength for fully compacted material	R	Required
RELV	Relative volume at which the material is fully compacted.	0.0 < R < 1.0	Required
TYPE	The type of data defined as the x-value in the tables.	C	CRUSH
	CRUSH Crush factor (1-relative volume)		
N	RELVOL Relative volume V/V_0.		

Field	Contents		Type	Default
OPTION	Material axes option used to determine how the local material axis system is defined.		C	ELEM
	VECT	Globally orthotropic with the material axes defined by two vectors V_1 and V_2, specified using the fields X1, Y1, Z1 and X2, Y2, Z2. The x-axis is defined by the vector V_1. The z-axis is defined as the cross product of V_1 and V_2. The y-axis is defined as the cross product of the z-axis and V_1.		
		 Material Axes Defined by Two Vectors		
	ELEM	Global orthotropic material with the material axes defined by element topology. The x-, y-, and z-axis are defined in the following way:		
		 Element Relative Grid Point Numbering		
TIDXX	Number of a TABLED1 entry defining the variation of the (local) xx-stress (y-value) with relative volume or crush (x-value).		I > 0	Required
TIDYY	Number of a TABLED1 entry defining the variation of the (local) yy-stress (y-value) with relative volume or crush (x-value).		I > 0	Required
TIDZZ	Number of aTABLED1 entry defining the variation of the (local) zz-stress (y-value) with relative volume or crush (x-value).		I > 0	Required

Field	Contents	Type	Default
TIDXY	Number of aTABLED1 entry defining the variation of the (local) xy-shear stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDYZ	Number of a TABLED1 entry defining the variation of the (local) yz-shear stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDZX	Number of a TABLED1 entry defining the variation of the (local) zx-shear stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDSR	Number of an optional TABLED1 entry defining the variation of a yield factor (y-value) with the deviatoric strain rate (x-value).	I > 0	See Remark 7.
EXX	The elastic modulus in the (local) x-direction when the material expands.	R > 0.0	Required
EYY	The elastic modulus in the (local) y-direction when the material expands.	R > 0.0	Required
EZZ	The elastic modulus in the (local) z-direction when the material expands.	R > 0.0	Required
GXY	The shear modulus in the (local) xy-direction when the material expands.	R > 0.0	Required
GYZ	The shear modulus in the (local) yz-direction when the material expands.	R > 0.0	Required
GZX	The shear modulus in the (local) zx-direction when the material expands.	R > 0.0	Required
BULKTYP	Bulk-viscosity model	C	DYNA
	DYNA Standard DYNA3D model		
BULKQ	Quadratic bulk-viscosity coefficient	R > 0.0	1.0
BULKL	Linear bulk-viscosity coefficient	R > 0.0	0.0
NUYX	The Poisson's ratio between the (local) x- and y-axis when the material expands.	-1.0 < R < 1.0	0.0
NUZX	Poisson's ratio between the (local) x- and z-axis when the material expands.	-1.0 < R < 1.0	0.0
NUZY	Poisson's ratio between the (local) y- and z-axis when the material expands.	-1.0 < R < 1.0	0.0
X1, Y1, Z1	Components of the vector V_1 in the basic coordinate system.	R	0.0
X2, Y2, Z2	Components of the vector V_2 in the basic coordinate system.	R	0.0

Remarks

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values are used.
2. If the initial Poisson's ratios are not supplied, the default is set to zero. Therefore, the behavior of the material during compaction is uncoupled. This means that straining in the (local) x-direction produces stresses only in the (local) x-direction, and not in the (local) y- or z-direction. The tables define the variation of the stress in a particular direction with the relative volume or the crush. The relative volume is defined as (current volume)/(initial volume) and varies from 1.0 (uncompressed) to 0.0 (zero volume). Crush is defined as one minus the relative volume and varies from 0.0 to 1.0. Since the tables should be defined with increasing x-values, it is convenient to use the default value for type, which is CRUSH. When defining the curves, care should be taken that the extrapolated values do not lead to negative yield stresses.
3. The elastic moduli (and the initial Poisson's ratios only if they are supplied) vary linearly with the relative volume from their initial uncompacted values to the fully compacted ones.
4. When the material is fully compacted, its behavior becomes isotropic with an elastic perfectly plastic material characteristic.
5. This material can only be used with Lagrangian solid elements.
6. If the TIDSR option is used, you can supply a table including strain-rate effects. Strain rate is defined here as the Euclidean norm of the deviatoric strain-rate tensor; i.e., $\dot{e} = \sqrt{\dot{e}_{ij}^{dev}\,\dot{e}_{ij}^{dev}}$. The y-values in this table are factors with which the stresses in the other tables are multiplied to incorporate strain-rate effects.
7. The behavior of this material is described in Materials.

ENDDATA

Terminates the Input Data

Marks the end of the input file.

Format and Example

1	2	3	4	5	6	7	8	9	10
ENDDATA									
ENDDATA									

Remarks

1. Anything after the ENDDATA entry is ignored.
2. An ENDDATA entry in an INCLUDE file is ignored.

EOSDEF

Deflagration

Defines the properties of the deflagration equation of state, and the reaction rate to model the burning of solid propellants. The burning of the solid propellant produces hot gas.

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSDEF	MID	GAMMA	B	R	C_v	C_p	E	RHOS	+EOSDEF1
EOSDEF	3	1.123	0.001	304			1E+6	1600	+EOSDEF1
+EOSDEF1	RHOF	W	BETA	SAVR	X	Y			
+EOSDEF1	0.1	4E-8	0.85	6000	0.6	0.0			

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
GAMMA	Constant γ	R>0.0	Required
B	Gas co-volume	R≥ 0.0	Required
R	Gas constant	R>0.0	Refer to remark 2.
CV	Specific heat at constant volume	R>0.0	Refer to remark 3.
CP	Specific heat at constant pressure	R>0.0	Refer to remark 3.
E	Chemical energy per unit burned mass	R>0.0	Required
RHOS	Reference density of propellant	R>0.0	Required
RHOF	$\frac{\text{Density of powder}}{\text{Reference density of propellant}}$	R>0.0	Refer to remark 5.
W	Burning rate coefficient	R≥ 0.0	Required
BETA	Burning rate exponent	R≥ 0.0	Required
SAVR	Initial surface area divided by volume	R>0.0	Required
X	Parameter form function	R≥ 0.0	Required
Y	Parameter form function	R≥ 0.0	Required

Remarks

1. Refer to EOSDEF-Deflagration (Ch. 4) in the *Dytran Theory Manual* for more details.
2. This equation of state is used with Eulerian elements.
3. The temperature of gas is calculated when one of the constants R, C_v or C_p is specified. When temperature is not mentioned in an output request, omit the constants.
4. The pressure in the reaction products is defined by the Noble-Abel equation of state as follows:

$p = (\gamma - 1)\frac{\rho}{1 - b\rho}e$ for reacted product,

$T = \frac{(\gamma - 1)e}{R}$ where γ, b are constants and R is the gas constant.

The chemical reaction rate for conversion of un-reacted explosive to reaction products is described by the following reaction rate equations:

$\xi = wSAVR$	vivacity
$\phi = (1 - F)^X + YF$	form function
$\frac{dF}{dt} = \xi\phi p^\beta$	time derivative of burn fraction

where

w =	burning rate coefficient
β =	burning rate exponent
$SAVR$ =	initial surface area divided by volume
Y =	parameter form function
S =	parameter form function

F denotes the burn fraction which is defined as the fraction of the explosive that has already reacted. For visualizing the burn fraction F, the Euler variable DEFBURN can be requested.

5. RHOF allows taking into account the air between gunpowder grains. Also, it allows taking into account air inside the combustion chamber. For example, if the chamber is filled with 10% real propellant and the rest is filled with air then RHOF = 0.1.
6. To indicate what Eulerian regions can burn, the variable DEFMAT can be used. In regions that can burn DEFMAT=1, DEFMAT can be set on the TICVAL card:
   TICVAL,14, DENSITY,1.14,SIE,300000.2,DEFMAT,1.0
7. Ignition of the propellant approached by setting SIE to the flame temperature in specific areas:
   TICVAL,15,DENSITY,1.14,SIE,4231908.591,DEFMAT,1.0

   Where $SIE_{ignition}$ = Flame Temp x Cv

Flame Temp can be extracted from Chemical Energy per unit burned mass and Gas constant: $T_{flame} = E / R$.

EOSEX

User-defined Equation of State

Defines an equation of state specified by a user subroutine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE mymat SCA.MDSolver.Obj.Uds.Dytran.Materials
```

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSEX	EID	NAME	VISC	GROUP					
EOSEX	12	WATER	0.01	mymat					

Field	Contents	Type	Default
EID	Unique equation of state number	I > 0	Required
NAME	Name of the equation of state passed to the user subroutine.	C	Blank
VISC	Viscosity coefficient	R > 0.0	No viscosity. See Remarks 4.and 5.
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. See Chapter 7: User Defined Services in the manual for a description of how to use user-written subroutines.
2. EID is passed to the usrMateos (cpp) or ext_mateos (Fortran) subroutines and can be used to identify the equation of state.
3. This entry can only be used for Lagrangian and Eulerian solids. Viscosity is only available for Eulerian solids.
4. The viscosity coefficient is the dynamic viscosity. It is the ratio between shear stress and velocity gradient. The SI-unit of viscosity is $Pa \cdot s = \frac{Ns}{m^2}$.
5. For the single mat solver viscous stresses can be requested by the use of TXX through TZX. Also, EFFSTS is available. For the multi-material solver viscous stresses are stored in TXX-VIS, TYY-VIS, TZZ-VIS, TXY-VIS, TYZ-VIS, TZX-VIS. These viscous stresses only depend on the current velocity gradients. The stresses like TXX are elastic-plastic stresses and depend on past stresses. The total stress tensor in the element is given by the average of the viscous stress and elastic-plastic stress. The weight factors are the material fraction of viscous fluid and the remaining materials.

EOSGAM — Gamma Law Gas Equation of State

Defines the properties of a Gamma Law equation of state where the pressure p is defined as

$$p = (\gamma - 1)\rho e$$

where

e	=	specific internal energy
ρ	=	overall material density
γ	=	a constant

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSGAM	EID	GAMMA	R	CV	CP	VISC			
EOSGAM	35	1.4							

Field	Contents	Type	Default
EID	Unique equation of state number	I > 0	Required
GAMMA	Constant γ	R ≥ 0.	Required
R	Gas constant	R > 0.0	See Remarks 2. and 3.
CV	Specific heat at constant volume	R > 0.0	See Remarks 2. and 3.
CP	Specific heat at constant pressure	R > 0.0	See Remarks 2. and 3.
VISC	Viscosity coefficient	R > 0.0	No viscosity. See Remarks 5.and 6.

Remarks

1. This equation of state is discussed in *Dytran Theory Manual*, Chapter 4: Models, EOSGAM – Gamma Law Equation of State.
2. The temperature of the gas is calculated when one of the gas constants, R, C_v, or C_p is specified. When temperature is not mentioned in an output request, the constants can be omitted.
3. The Euler variable name for temperature is TEMPTURE.
4. Gamma, R, C_v, and C_p have the following relationships:

$$\gamma = \frac{C_p}{C_v} \quad R = C_p - C_v$$

5. The viscosity coefficient is the dynamic viscosity. It is the ratio between shear stress and velocity gradient. The SI-unit of viscosity is $Pa \cdot s = \frac{Ns}{M^2} = \frac{kg}{ms}$.
6. If possible, use (in coupled analysis) the FASTCOUP coupling algorithm because viscous fluxes are computed more accurately for fast coupling than for general coupling.
7. For the single mat solver, viscous stresses can be requested by the use of `TXX` through `TZX`. Also, `EFFSTS` is available. For the multi-material solver, viscous stresses are stored in `TXX-VIS`, `TYY-VIS`, `TZZ-VIS`, `TXY-VIS`, `TYZ-VIS`, `TZX-VIS`. These viscous stresses depend only on the current velocity gradients. The stresses like `TXX` are elastic-plastic stresses and depend on past stresses. The total stress tensor in the element is given by the average of the viscous stress and elastic-plastic stress. The weight factors are the material fraction of viscous fluid and the remaining materials.

EOSIG — Ignition and Growth Equation of State

Defines the properties of Ignition and Growth equation of state and the reaction rate equation used to model high explosives.

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSIG	EID	AE	BE	R1E	R2E	OMGE	I	G	+EOIG1
EOSIG	30								+EOIG1
+EOIG1	A	AP	BP	R1P	R2P	OMGP	X	Y	+EOIG2
+EOIG1									+EOIG2
+EOIG2	Z	R	ECHEM	PRSTOL	ITRMAX	UNITDEF	DBEXPL	UNITCNV	
+EOIG2									

Field	Contents	Type	Default
EID	Unique equation of state number	I > 0	Required
AE	Constant A_e for un-reacted explosive	R	Required. See Remark 5.
BE	Constant B_e for un-reacted explosive	R	Required. See Remark 5.
R1E	Constant R_{1e} for un-reacted explosive	R	Required. See Remark 5.
R2E	Constant R_{2e} for un-reacted explosive	R	Required. See Remark 5.
OMGE	Constant ω_e for un-reacted explosive	R	Required. See Remark 5.
I	First ignition coefficient	R	Required. See Remark 5.
G	Second ignition coefficient	R	Required. See Remark 5.
A	Density ignition coefficient	R	Required; See Remark 5.
AP	Constant A_p for reacted product	R	Required. See Remark 5.
BP	Constant B_p for reacted product	R	Required. See Remark 5.
R1P	Constant R_{1p} for reacted product	R	Required. See Remark 5.
R2P	Constant R_{2p} for reacted product	R	Required. See Remark 5.
OMGP	Constant ω_p for reacted product	R	Required. See Remark 5.
X	Surface burning exponent	R	2./9; see Remark 5.
Y	Surface burning exponent	R	2./3; see Remark 5.
Z	Pressure exponent	R	Required. See Remark 5.
R	Relative density exponent	R	4; see Remark 5.
ECHEM	Chemical energy of high explosive per unit mass	R	Required. See Remark 5.

Field	Contents		Type	Default
PRSTOL	Tolerance for pressure equilibrium iterations in mixed phase elements		R>0.0	1.E-6
ITRMAX	Maximum number of iterations in pressure equilibrium iterations		I>0	16
UNITDEF	User-defined default unit for the inputs:		C	See Remarks 2. and 3.
	CGMS	cm/g/μs units		
	SI	International System units		
	METRIC	Metric units		
	IMPER	Imperial units		
	MMMGS	mm/mg/μs units		
DBEXPL	Use explosive material from the database (See Remarks 4. and 6.). The following detonation materials are available in the data base:		C	NO
	NO	The database is not used. See Remark 5.		
	P94A	PBX-9404 (a) explosive		
	TATB	TATB explosive		
	PETN	PETN explosive		
	CTNT	Cast TNT explosive		
	LCOMPB	LANL COMP B explosive		
	MCOMPB	Military COMP B explosive		
	P94B	PBX-9404 (b) explosive		
	LX17	LX-17 explosive		
UNITCNV	User-defined conversion units:		C	See Remarks 2.and 3.
	CGMS	cm/g/μs units		
	SI	International System units		
	METRIC	Metric units		
	IMPER	Imperial units		
	MMMGS	mm/mg/μs units		

Remarks

1. This equation of state can be used with solid Lagrangian and Eulerian elements.
2. The definition of the unit system in which the input values are defined is required information only in case you wish to have an automatic conversion to a different unit system as defined by the UNITCNV field. In case you are using the conversion mechanism, note that the density RHO in the corresponding DMAT entry is interpreted in the unit system defined here.

Table 5-1 defines sets of units available:

Table 5-1 Sets of Units used in the IG Model

Quantity	CGMS	SI	METRIC	IMPERIAL	MMMGS
Length	Centimeter (cm)	Meter (m)	Centimeter (cm)	Inch (in)	Millimeter (mm)
Time	Microsecond (μs)	Second (s)	Second (s)	Second (s)	Microsecond (μs)
Mass	Gram (g)	Kilogram (kg)	Gram (g)	Slug (lbf-s2/in)	Milligram (mg)
Force	Teradyne	Newton (N)	Dyne	Pound force (lbf)	kN
Density	g/cm3	Kg/m3	g/cm3	lbf-s2/in4	mg/ mm3
Stress	Mbar	Pascal (Pa)	μbar	Lbf/in2	GPa
Energy	1012 erg (Mbars-cm3)	Joule (J)	Erg	Lbf-in	J
Temperature	Kelvin (K)	Kelvin (K)	Kelvin (K)	Kelvin (K)	Kelvin (K)

3. The UNITCNV field defines the unit system to which the material parameters are converted. In case you are not using one of the database material models, you also have to define the default unit system (UNITDEF) in which you supplied the data.
4. You can use the database containing several detonation materials to start the analysis. The material data are taken from Lee/Tarver (Ref.1.) and Murphy/Lee (Ref.2.) papers in the *Dytran Theory Manual*. The equations of state parameters are given in the table 2.
5. The default setting for DBEXPL is NO, which means you should define the values in the input fields (fields 3 to 18). If the database material name is defined, all values in the input fields will be overridden. The reference density RHO defined on the corresponding DMAT entry is set to the value from the database.
6. The default unit system for the material database parameters is the CGMS unit system. If you wish to use the material base data in a different unit system, you can specify this by defining the target unit system in the UNITCNV field.
7. You can define the shear property and yield model of the material with respectively SHXXX and YLDXX entry. Note that the unit system of data required in these entries should be consistent with the unit system defined in the UNITCNV field.
8. The IG equation of state cannot be used in combination with a spallation model.

9. The following JWL equation of state is used to calculate the pressure of the un-reacted explosive (in "solid" state):

$$p_e = A_e\left(1-\frac{\omega_e\eta_e}{R_{1e}}\right)e^{\frac{-R_{1e}}{\eta_e}} + B_e\left(1-\frac{\omega_e\eta_e}{R_{2e}}\right)e^{\frac{-R_{2e}}{\eta_e}} + \omega_e\eta_e\rho_0 E_e$$

where

$\eta_e = \frac{\rho_e}{\rho_0}$	=	the relative density of the un-reacted explosive
E_e	=	the specified internal energy per unit mass of the unreacted explosive
ρ_0	=	the initial density of the explosive
$A_e, B_e, \omega_e, R_{1e}, R_{2e}$	=	the input constants of the un-reacted explosive.

Similarly, the pressure in the reaction products (in "gas" state) is defined by another JWL form as follows:

$$p_p = A_p\left(1-\frac{\omega_p\eta_p}{R_{1p}}\right)e^{\frac{-R_{1p}}{\eta_p}} + B_p\left(1-\frac{\omega_p\eta_p}{R_{2p}}\right)e^{\frac{-R_{2p}}{\eta_p}} + \omega_p\eta_p\rho_0 E_p$$

where

$\eta_p = \frac{\rho_p}{\rho_0}$	=	the relative density of the reaction product.
E_p	=	the specified internal energy per unit mass of the reacted product.
$A_p, B_p, \omega_p, R_{1p}, R_{2p}$	=	the input constants of the reaction product.

The chemical reaction rate for conversion of un-reacted explosive to reaction products is described by the following reaction rate equation:

$$\frac{\partial F}{\partial t} = I(1-F)^x(\eta_e - 1 - a)^y + G(1-F)^x F^y (P)^z$$

Here F denotes the burn fraction that is defined as the fraction of the explosive that has already reacted. For more details concerning the implementation of this equation of state, please refer to the *Dytran Theory Manual*.

10. You can access the results of the un-reacted explosive and reaction products for IG elements. These EOSIG specific output variables are:

Keyword	Description
SIE-E	Specific internal energy of un-reacted explosive part
SIE-P	Specific internal energy of reaction products part
FMAT	Volume fraction
RHO-E	Density of un-reacted explosive part
RHO-P	Density of reaction products part
MASS-E	Mass of un-reacted explosive part
MASS-P	Mass of reaction products part

The output variables for the burn fraction are

Keyword	Type of Elements	Description
FBURN	Solid Lagrangian Elements	Burn fraction of EOSIG material
	Euler Elements	Not applicable for EOSIG materials. Burn fraction for EOSJWL material
IGBURN	Solid Lagrangian Elements	Not available
	Euler Elements	Burn fraction of EOSIG MATERIAL

11. The ignition of IG material can be initiated by:
    - Compression of the IG material in a small region, where the compression originates from outside that region. This is the most physical method to initiate ignition. Examples are a shock wave entering the region, a flow boundary that supplies mass to the region and a plate or other structural part that compresses the region. In all these cases the IG material should be initialized with zero pressures. This can be achieved by not specifying the specific energy on the TICVAL entry that prescribes the initial state of the IG material. The specific energy will be computed such that the initial pressure is zero.
    - Compression of the IG material in a small region where the compression originates within that region. This can be done by specifying either a density that exceeds the compression limit or a specific energy that gives rise to a sufficiently large pressure.

Table 5-2 Coefficients for the IG Model of Several Explosions in the Database

Explosive	PBX-9404 (a)	TATB	PETN	Cast TNT	LANL COMP B	Military COMP B	PBX-9404 (b)	LX-17
Un-reacted Equation of State and Constitutive Values:								
RHO (g/cm3)	1.842	1.90	1.842	1.61	1.712	1.630	1.842	1.903
AE (μbar)	69.69	108.2	37.46	17.98	778.1	1479.	9522.	778.1
BE (μbar)	-1.727	-2.406	-1.313	-0.931	-0.05031	-0.05261	-0.5944	-0.05031
R1E	7.8	8.2	7.2	6.2	11.3	12.	14.1	11.3
R2E	3.9	4.1	3.6	3.1	1.13	1.2	1.41	1.13
OMGE	0.8578	1.251	1.173	0.8926	0.8938	0.9120	0.8867	0.8938
Reacted Product Equation of State Values:								
AP (μbar)	8.524	6.5467	6.17	3.712	5.242	5.5748	8.524	6.5467
BP (μbar)	0.1802	0.071236	0.16926	0.032306	0.07678	0.0783	0.1802	0.071236
R1P	4.6	4.45	4.4	4.15	4.2	4.5	4.6	4.45
R2P	1.3	1.2	1.2	0.95	1.1	1.2	1.3	1.2
OMGP	0.38	0.35	0.25	0.30	0.34	0.34	0.38	0.35
ECHEM μbar-cm3/g)	0.0554	0.0363	0.0548	0.0435	0.0496	0.04969	0.0554	0.03626
Reaction Rate Parameters:								
I (μs-1)	44.0	50.0	20.0	50.0	44.0	44.0	44.0	50.0
G (μbar-z μs-1)	200.0	125.0	400.0	40.0	414.0	514.0	850.0	500.0
A	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0
Z	1.6	2.0	1.4	1.2	2.0	2.0	2.0	3.0
X	2/9	2/9	2/9	2/9	2/9	2/9	2/9	2/9
Y	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3
R	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0

EOSJWL

JWL Explosive Equation of State

Defines the properties of a JWL equation of state commonly used to calculate the pressure p of the detonation products of high explosives

$$p = p_0 + A\left(1 - \frac{\omega\eta}{R_1}\right)e^{\frac{-R_1}{\eta}} + B\left(1 - \frac{\omega\eta}{R_2}\right)e^{\frac{-R_2}{\eta}} + \omega\eta\rho_0 e + \frac{\omega\rho\lambda Q}{\rho_0}$$

$$\frac{d\lambda}{dt} = a(1-\lambda)^m p^n$$

e	=	specific internal energy
ρ_0	=	reference density
ρ	=	overall material density
η	=	ρ/ρ_0
p_0	=	initial pressure
λ	=	fraction that describes the afterburning. It ranges from 0 to 1
A, B, R_1, R_2, a, m, n and Q are constants.		

The last term: $\frac{\omega\rho\lambda Q}{\rho_0}$ models afterburning.

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSJWL	MID	A	B	R1	R2	OMEGA	P0	Q	+EOSJWL1
EOSJWL	37	5.2E11	0.77E11	4.1	1.1	0.34		0.0	+EOSJWL1
+EOSJWL1	a	m	n						
+EOSJWL1	0.0	0.0	0.0						

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
A	Constant A.	R	0.0
B	Constant B.	R	0.0
R1	Constant R_1.	R	0.0
R2	Constant R_2.	R	0.0
OMEGA	Constant ω.	R	0.0
P0	Initial pressure	R	0.0; see Remark 4.
Q	Initial pressure	R≥ 0.0	0.0; see Remark 6.
a	Initial pressure	R≥ 0.0	0.0; see Remark 6.
m	Initial pressure	R≥ 0.0	0.0; see Remark 6.
n	Initial pressure	R≥ 0.0	0.0; see Remark 6.

Remarks

1. This equation of state can be used only with Eulerian elements.
2. A DETSPH entry must be used to specify the detonation model.
3. This equation of state is discussed in *Dytran Theory Manual*, Chapter 4: Models, EOSJWL – JWL Equation of State.
4. In simulations with ideal gases, the JWL material needs to have an initial pressure to counter balance the pressure of the ideal gas. Similarly, in case of under calculations where an explosive is located at a certain depth, P0 should be preset to equal the initial hydrostatic pressure.
5. The DETSPH definition of one EOSJWL material also applies to all other EOSJWL materials. Therefore, the blast wave of one explosive can ignite other explosives. To prevent this linked ignition, PARAM, JWLDET, NOLINK can be used.
6. By default the constants *a,m,n* and *Q* are zero and afterburning is not taken into account. To model afterburning all the variables *Q, a, m* and *n* have to be set to positive constants. The extent of the afterburning can be checked by requesting the Euler element variable AFTERBURN. This variable equals λ. If the afterburning in an Euler element is complete then AFTERBURN = 1.0.

EOSMG — Mie-Gruneisen Equation of State

Defines the properties of a Mie-Gruneisen equation of state commonly used to calculate the pressure p in high strain rate processes.

$$p = \frac{\rho_0 c^2 \eta}{(1-s\eta)^2}\left(1-\frac{\Gamma_0 \eta}{2}\right) + \Gamma_0 \rho_0 e$$

$$\eta = 1-\frac{\rho_0}{\rho_1}$$

$$\rho_1 = min(\rho, RM)$$

where

e	=	specific internal energy. For material at zero pressure, e has to be initialized as zero.
ρ_0	=	reference density
ρ	=	overall material density
Γ_0	=	Gruneisen parameter at reference density
s	=	definition by $U_s = c_0 + sU_p$ where U_s and U_p are the linear shock velocity and particle velocity, respectively, as obtained from the shock data.
c	=	sound speed at reference density
RM	=	cut-off value for density

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSMG	EID	C	S	Γ_0	RM				
EOSMG	37	2000	1.5	2.0	2000				

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
C	Sound speed at reference density.	R	Required
S	Constant s.	R	Required
Γ_0	Gruneisen gamma	R	Required
RM	Cut-off value for density.	R	Required

Remarks

1. This equation of state can only be used with Eulerian elements.
2. This equation of state is discussed in *Dytran Theory Manual*, Chapter 4: Models, EOSMG - Mie-Gruneisen Equation of State.
3. The cut off value RM is only used for limiting the pressure. To prevent division by zero, RM should be less than $\frac{s}{s-1}\rho_{ref}$. RM can be set slightly below this value. In case the simulation gets unstable because of too large pressures, RM can be decreased.

EOSNA Noble-Abel equation of state

The EOSNA defines an equation of state based on the Noble-Abel gas law. This is an adjustment of the ideal gas law that takes into account the volume of gas molecules. Where pressure p is defined as follows:

$$p = (\gamma - 1)\frac{\rho}{1 - b\rho}e$$

where:

e	=	specific internal energy
ρ	=	overall material density
γ	=	a constant
b	=	Gas co-volume

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSNA	EID	GAMMA	B	R	C_V	C_P			
EOSNA	35	1.2363	0.001	314.46					

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
GAMMA	Constant Υ	R≥ 0.0	Required
B	Gas co-volume	R≥ 0.0	0.0 Refer to Remark 3.
R	Gas constant	R>0.0	Refer to Remark 4.
CV	Specific heat at constant volume	R>0.0	Refer to Remark 4.
CP	Specific heat at constant pressure	R>0.0	Refer to Remark 4.

Remarks

1. Refer to EOSNA – Noble-Abel equation of state, Chapter 4-Models of *Dytran Theory Manual*.
2. EOSNA – Noble-Abel equation of state is used with Eulerian elements only.
3. The parameter B models the interactions between gas particles.
4. The temperature of the gas is calculated when one of the gas constants, R, C_v, and C_p is mentioned.
5. The Euler variable name for temperature is TEMPTURE.
6. The relation between γ, R, C_v, and C_p is shown below:

$$\gamma = \frac{C_p}{C_v} \quad R = C_p - C_v$$

EOSPOL

Polynomial Equation of State

Defines the properties of a polynomial equation of state where the pressure p is defined as follows:

In compression $(\mu > 0)$,

$$p = a_1\mu + a_2\mu^2 + a_3\mu^3 + (b_0 + b_1 + b_2\mu^2 + b_3\mu^3)\rho_0 e$$

In tension $(i < 0)$,

$$p = a_1\mu + (b_0 + b_1\mu)\rho_0 e$$

where

μ	=	$\eta - 1$
η	=	ρ/ρ_0
ρ	=	overall material density
ρ_0	=	reference density
E	=	specific internal energy

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSPOL	EID	A1	A2	A3	B0	B1	B2	B3	+
EOSPOL	100	80.E6							+
+	HVL	VISC							
+	1.1								

Field	Contents	Type	Default
EID	Unique equation of state number	I > 0	Required
A1	Coefficient a1 or Bulk Modulus	R	0.0
A2	Coefficient a2	R	0.0
A3	Coefficient *a3*	R	0.0
B0	Coefficient b0	R	0.0
B1	Coefficient b1	R	0.0
B2	Coefficient b2	R	0.0
B3	Coefficient b3		R

Field	Contents	Type	Default
HVL	Hydrodynamic volume limit	R ≥ 1.0	1.1
VISC	Viscosity coefficient	R > 0.0	No viscosity. See Remarks 4. and 5.

Remarks

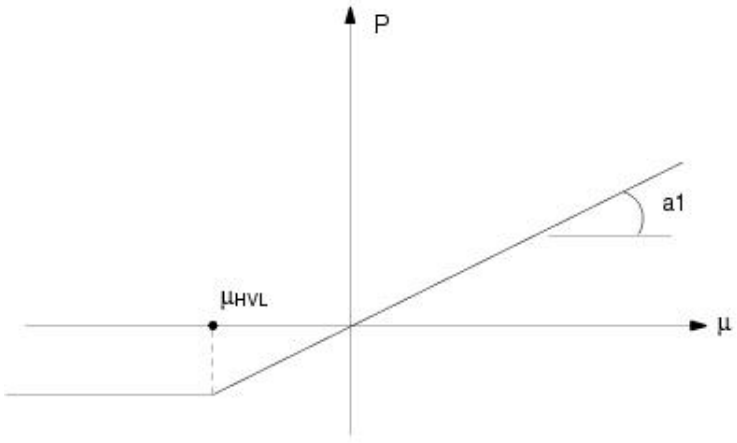
1. When the relative volume (ρ_0/ρ) exceeds HVL, the pressure is cut off to

   $$P_{HVL} = f(\mu_{HVL})$$

   with

   $$\mu_{HVL} = \frac{1}{HVL} - 1$$

   e.g., for $p = a1 \cdot \mu$, the pressure behavior is as follows:



2. When the PARAM,HVLFAIL is set to YES, the elements where the relative volume (ρ_0/ρ) exceeds HVL fail completely. Their stress state is zero.
3. This equation of state is discussed in *Dytran Theory Manual*, Chapter 4: Models, EOSPOL – Polynomial Equation of State.
4. The viscosity coefficient is the dynamic viscosity. It is the ratio between shear stress and velocity gradient. The SI-unit of viscosity is $Pa \cdot s = \frac{Ns}{M^2} = \frac{kg}{ms}$.
5. If possible, use in coupled analysis the FASTCOUP coupling algorithm, because viscous fluxes are computed more accurately for fast coupling than for general coupling.

6. For the single mat solver, viscous stresses can be requested by the use of TXX through TZX. Also EFFSTS is available. For the multi-material solver, viscous stresses are stored in TXX-VIS, TYY-VIS, TZZ-VIS, TXY-VIS, TYZ-VIS, TZX-VIS. These viscous stresses only depend on the current velocity gradients. Stresses like TXX are elastic-plastic stresses and depend on past stresses. The total stress tensor in the element is given by the average of the viscous stress and elastic-plastic stress. The weight factors are the material fraction of viscous fluid and the remaining materials.

EOSTAIT

Tait Equation of State

Defines the properties of an equation of state based on the Tait model in combination with a cavitation model where the pressure p is defined as follows:

No cavitation $(\rho > \rho_e)$,

$$p = a_0 + a_1(\eta^\gamma - 1)$$

Cavitation $\rho \le \rho_c$,

$$p = p_c$$

where

η	=	ρ/ρ_0
ρ	=	overall material density
ρ_0	=	reference density
ρ_c	=	critical density which produces the cavitation pressure p_c

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSTAIT	EID	A0	A1	GAMMA	RHOC	VISC			
EOSTAIT	3	1.E6	3.31E9	7.15	.9999578	.0001			

Field	Contents	Type	Default
EID	Unique equation of state number	I > 0	Required
A0	Constant a_0	R	0.0
A1	Constant a_1	R	0.0
GAMMA	Constant η	R > 0.0	1.0
RHOC	Constant ρ_c	R	Required
VISC	Viscosity coefficient	R > 0.0	No viscosity. See Remarks 4.and 5.

Remarks

1. The pressure can not fall below the cavitation pressure $p_c = a_0 + a_1\left(\left(\frac{\rho_c}{\rho_0}\right)^{\gamma} - 1\right)$, although the density can continue to decrease below its critical value ρ_c.
2. The Tait equation of state can not be used in combination with a spallation model.
3. For a more detailed description, see the *Dytran Theory Manual*, Chapter 4: Models, EOSTAIT – Tait Equation of State.
4. The viscosity coefficient is the dynamic viscosity. It is the ratio between shear stress and velocity gradient. The SI-unit of viscosity is $Pa \cdot s = \frac{Ns}{M^2} = \frac{kg}{ms}$.
5. If possible, use in coupled analysis the FASTCOUP coupling algorithm, because viscous fluxes are computed more accurately for fast coupling than for general coupling.
6. For the single mat solver, viscous stresses can be requested by the use of `TXX` through `TZX`. Also `EFFSTS` is available. For the multi-material solver, viscous stresses are stored in `TXX-VIS`, `TYY-VIS`, `TZZ-VIS`, `TXY-VIS`, `TYZ-VIS`, `TZX-VIS`. These viscous stresses only depend on the current velocity gradients. Stresses like `TXX` are elastic-plastic stresses and depend on past stresses. The total stress tensor in the element is given by the average of the viscous stress and elastic-plastic stress. The weight factors are the material fraction of viscous fluid and the remaining materials.

FABRIC

Woven Fabric Material

Defines the properties of a bi-directional woven fabric material for shell elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
FABRIC	MID	RHO	ECOAT	NUCOAT	GCOAT	DAMPCOAT	COMPCOAT	PERC	+
FABRIC	3	850.	5.52E6	0.33				50.	+
+	E1L	E1Q		THETA1	XWARP	YWARP	ZWARP		+
+	21.6E7				1.0	0.0	0.0		+
+	E2L	E2Q		THETA2	XWEFT	YWEFT	ZWEFT		+
+	21.6E7				0.0	1.0	0.0		+
+	SCOF	G12	DAMPFIB	COMPFIB	LOCKANG1	LOCKANG2			
+									

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
RHO	Density	R > 0.0	Required
ECOAT	Young's modulus of coating material	R > 0.0	See Remark 2.
NUCOAT	Poisson's ratio of coating material	R > 0.0	See Remark 2.
GCOAT	Shear modulus of coating material	R > 0.0	See Remark 2.
DAMPCOAT	Damping is applied to the coating stresses: $d\sigma_{ij} = DAMPCO \cdot E \cdot \varepsilon_{ij} \cdot dt_{elm}$	R ≥ 0.0	0.1 See Remark 3.
COMPCOAT	Scale factor for coating compression stresses	1.0≥ R ≥ 0.0	1.0 See Remark 4.
PERC	Thickness percentage of coating material	100.0≥ R ≥ 0.0	0.0 (no coating)
E1L	Young's modulus of fabric in warp direction, linear coefficient	R > 0.0	Required
E1Q	Young's modulus of fabric in warp direction, quadratic coefficient	R ≥ 0.0	0.0
THETA1	Orientation angle between the element coordinate system and the warp ends	R	See Remark 5.

Field	Contents	Type	Default
XWARP, YWARP, ZWARP	Vector indicating the warp direction of the fabric material. The vector is with respect to the basic coordinate system.	R	(1., 0., 0.) See Remark 5.
E2L	Young's modulus of fabric in weft direction, linear coefficient	R > 0.0	Required
E2Q	Young's modulus of fabric in weft direction, quadratic coefficient	R ≥ 0.0	0.0
THETA2	Orientation angle between the element coordinate system and the weft ends	R	See Remark 5.
XWEFT, YWEFT, ZWEFT	Vector indicating the weft direction of the fabric material. The vector is with respect to the basic coordinate system.	R	(0., 1., 0.) See Remark 5.
SCOF	Shear coefficient of friction	R	0.0 See Remark 7.
G12	Shear modulus of fabric material	R	See Remark 8.
DAMPFIB	Damping is applied to the fiber stresses: $d\sigma_{ij} = DAMPFI \cdot E \cdot \varepsilon_{ij} \cdot dt_{elm}$	R ≥ 0.0	0.1 See Remark 3.
COMPFIB	Scale factor for fiber compression stresses	1.0≥ R ≥ 0.0	1.0 See Remark 4.
LOCKANG1	Locking angle 1 for change in fiber cross-over angle	R ≥ 0.0	10.0 See Remark 11.
LOCKANG2	Locking angle 2 for change in fiber cross-over angle	R ≥ 0.0	15.0 See Remark 11.

Remarks

1. For a description of the bi-directional woven fabric model, see *Dytran Theory Manual*, Chapter 3: Materials.
2. When a coating is defined (PERC>0), two out of three values need to be specified for ECOAT, NUCOAT, and GCOAT.
3. For air bag modeling the following values of DAMPCOAT and DAMPFIB are suggested:

   ```
   DAMPCOAT= 0.05
   DAMPFIB= 0.05
   ```

4. The compressive stresses in the fibers are scaled with the value of COMPFIB. Putting COMPFIB = 0.0 results in a tension only fiber model.

   The compressive stresses in the coating are scaled with the value of COMPCOAT. Putting COMPCOAT = 0.0 results in a tension only coating model.

   The compressive stresses are scaled in the direction of the principal stresses.

When PERC = 100%, and the coating of this fabric model is used to simulate an isotropic air bag material, it is best to scale down the compressive stresses of the coating. A suggested value is COMPCOAT = 0.1.

5. Since this is a model which tracks warp and weft directions and uses total warp/weft strain as a state variable, the initial warp and weft directions must be specified. There are two ways to indicate the initial warp and weft directions:
   a. THETA1 and THETA2

      Orientation angles between the element coordinate system and the warp/weft ends. If no orientation angle is specified, vectors will be used to indicate the warp/weft directions of the fabric material with respect to *the basic coordinate system*.

   b. XWARP, YWARP, ZWARP and XWEFT, YWEFT, ZWEFT

      Vectors indicating the warp/weft directions of the fabric material with respect to *the basic coordinate system*. The projection of these vectors on the surface of each element is used to determine the angle between the element and the material coordinate system. If the orientation angles are defined, these vectors are ignored.
6. For shell element properties (PSHELL1), when the material is FABRIC, the material angle THETA is ignored. The orientation of the fabric fibers is defined completely on the FABRIC entry.

   For layered composite element properties (PCOMP), when the material of a ply is FABRIC, the angle THETAi is ignored. The orientation of the fabric fibers is defined completely on the FABRIC entry.
7. The maximum shear stress is given by a friction coefficient of the fabric (SCOF) times the RMS value of the direct fiber stresses.
8. If the field G12 is left blank, the shear modulus is computed from the RMS value of the two linear stiffness coefficients.
9. When Fabric material is referenced by shell elements, the Spin Rate method (SPIN) is applied automatically when no stress-rotation correction is specified on SPINCOR option. See PSHELL entry for the details on SPINCOR option.
10. There are a number of specific output sublayer variables useful for this material:

Q1AF	Direction cosines/sines between the element coordinate
Q2AFIB	System and the warp ends
Q1BFIB	Direction cosines/sines between the element coordinate
Q2BFIB	System and the weft picks
SGMA	Direct stress in fabric parallel to the warp ends
SGMB	Direct stress in fabric parallel to the weft picks
SGFRIC	Stress due only to shear in the weave of the fabric
EPSFA	Strain in fabric parallel to the warp ends
EPSFB	Strain in fabric parallel to the weft picks
ANGLE	Crossover angle between warp ends and weft picks

11. When a fabric is being sheared, the angle between the fibers changes. At a certain moment, the fibers will reach a locking angle, after which a further change in the fiber angle is no longer possible.

    The simulation models this behavior as follows:

    a. Change in Fiber Crossover Angle < LockAng1

       The shear stress between the fibers is cut off based on the friction coefficient `SCOF`

    b. LockAng1 < Change in Fiber Crossover Angle < LockAng2

       The shear stress between the fibers is linearly increased.

    c. Change in Fiber Crossover Angle > LockAng2

       The shear stress between the fibers is no longer cut off.

       This situation is equal to an infinite friction coefficient `SCOF`.

FAILEST Maximum Equivalent Stress and Minimum Time Step Failure Model

Defines the properties of a failure model where total failure occurs when the equivalent stress exceeds the specified value and the element time step drops below the specified limit.

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILEST	FID	MES	DT						
FAILEST	1	1.E9	1.E-9						

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
MES	Maximum equivalent stress that causes failure on the deviatoric part of the stress tensor.	R	Required
DT	Minimum time step that causes total failure.	R	Required

Remarks

1. This failure model is valid for Lagrangian solid (CHEXA) orthotropic materials. (See also the DMATOR entry.)
2. The FAILEST failure model is a two-stage failure. The first stage retains the hydrodynamic properties of the material. The second stage is reached when the global time step falls below the specified value. The element is then removed from the calculation.

FAILEX

User Failure Subroutine

Specifies that a user subroutine is being used to define the failure model.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE usrfail SCA.MDSolver.Obj.Uds.Dytran.Materials
```

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILEX	FID	GROUP							
FAILEX	200	usrfail							

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual.

FAILEX1

Extended User Failure Subroutine

Specifies that a user subroutine is being used to define a failure model.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE usrfail SCA.MDSolver.Obj.Uds.Dytran.Materials
```

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILEX1	FID	GROUP							
FAILEX1	300	usrfail							

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual.
2. The failure model is available for orthotropic materials only. The FAILEX1 entry must be referenced on the DMATOR entry.
3. The failure model allows for an extensive description of the failure of composite materials in three-dimensional elements. It includes the possibility to have property degradation according to material damage.

FAILEX2

User Failure Subroutine

Defines a damage model specified by a user subroutine. In addition, the VOLPLS and SOFTE element variables can be set by this routine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE usrfail SCA.MDSolver.Obj.Uds.Dytran.Materials
```

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILEX2	FID	METH	GROUP						
FAILEX2	200	CONT	usrfail						

Field	Contents		Type	Default
FID	Unique failure model number		I > 0	Required
MTH	Specifies how failure is applied		C	CONT
	CONT	continuous failure		
	DISC	discrete failure		
	NOFAIL	No failure		
GROUP	The group name used for the FMS section CONNECT SERVICE statement		C	Required

Remarks

1. This model is only supported by the Multi-material solver with strength.
2. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual.
3. For each material and for each Euler element a variable will be created that monitors the degree of failure of the material. This variable is denoted by DAMAGE and is between 0 and 1. The usrMatfail2 (cpp) and ext_fail2 (Fortran) user subroutines allow updating this damage variable due to the plastic strain increment of the current cycle.
4. There are two ways in which this damage variable can model failure These are:
   - Continuous failure: The yield stress is reduced by a factor (1-D). When D exceeds 1, the yield stress equals zero and the element fails.
   - Discrete failure: the element fails when *D* equals one.

   When NOFAIL is set, positive damage values will not lead to failure. This is useful if the failure modeling is done by usrMatyld(cpp) and ext_matyld (Fortran) user routines. Then the yield stress can be reduced depending on the magnitude of the damage variable.

FAILJC

Johnson-Cook Failure Model

Defines the properties of a failure model where failure is determined by a damage model. The damage model is given by:

$$D = \sum_{time} \frac{\Delta \varepsilon_p}{\varepsilon^{frac}}$$

$$\varepsilon^{frac} = (D_1 + D_2 exp(D_3 \sigma^*)) \left(1 + D_4 ln \frac{\dot{\varepsilon}_{pl}}{\dot{\varepsilon}^0_{pl}}\right) (1 + D_5 T^*)$$

$$\sigma^* = \frac{\sigma_m}{\bar{\sigma}}$$

$$T^* = \frac{T - T_{room}}{T_{melt} - T_{room}}$$

The summation is performed over all past time increments. The variable D measures the damage; T is the temperature, σ_m the mean stress, $\bar{\sigma}$ is the von Mises equivalent stress, and ε^{frac} is the fracture strain. The fracture strain depends on a nondimensional plastic strain rate $\dot{\varepsilon}_{pl}/\dot{\varepsilon}^0_{pl}$. If D exceeds one it set equal to one.

The damage variable D is transported along with the Eulerian material.

There are two methods to determine when elements fail:

- Continuous failure: The yield stress is reduced by a factor $(1 - D)$. When D exceeds 1 the yield stress equals zero and the element fails.
- Discrete failure: the element fails when D equals one.

This failure model applies to high-strain rate deformation of metals. It is less suitable for quasi-static problems.

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILJC	FID	D1	D2	D3	D4	D5	ELPLDOTREF	TROOM	+
FAILJC	1	.05	3.44	-2.1	0.002	0.61	1.0	0	+
+	TMELT	CP	MTH						
+	1495	450	CONT						

Field	Contents		Type	Default
FID	Unique failure model number		I > 0	Required
D1..D5	Parameters		R	0.0
ELPLDOTREF	Reference plastic strain rate		R	1.0
TROOM	Room temperature		R	293
TMELT	Melt temperature		R	1.E+20
CP	Heat capacity		R	1.E+20
MTH	Specifies how failure is applied:		C	CONT
	CONT	continuous failure		
	DISC	discrete failure		
	NOFAIL	damage is not used for failure. Failure modeling can then be done by an EXYLD subroutine.		

Remarks

1. This failure model is only available for the materials of Eulerian elements and Lagrangian solid elements. The use of the multi-material solver with strength is required.
2. The variable D can be visualized by adding DAMAGE to the Output request for Euler elements and Lagrangian solid elements.

FAILMES

Maximum Equivalent Stress Failure Model

Defines the properties of a failure model where failure occurs when the equivalent stress exceeds the specified value.

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILMES	FID	MES							
FAILMES	1	1.E9							

Field	Contents	Type	Default
FID	Unique failure model number	I > 0	Required
MES	Maximum equivalent stress that causes failure	R	Required

Remark

This failure model is valid for Lagrangian solid element materials. (See also the DMAT and DMATOR entries.)

FAILMPS — Maximum Plastic Strain Failure Model

Defines the properties of a failure model where failure occurs when the equivalent plastic strain exceeds the specified value.

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILMPS	FID	MPS	MPS-C						
FAILMPS	1	.15							

Field	Contents	Type	Default
FID	Unique failure model number	I > 0	Required
MPS	Maximum plastic strain that causes failure	R	Required
MPS-C	Maximum plastic strain when material is under compression that causes failure	R	MPS

Remark

This failure model is valid for Eulerian, shell (CQUAD4 and CTRIA3), Hughes-Liu beams, and Lagrangian solid element materials. (Also see the DMAT and DMATEP entries.)

FAILPRS

Maximum Pressure Failure Model

Defines the properties of a failure model where failure occurs when the hydrodynamic pressure exceeds the specified value.

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILPRS	FID	PRS							
FAILPRS	1	5.E8							

Field	Contents	Type	Default
FID	Unique failure model number	I > 0	Required
PRS	Maximum pressure that causes failure	R	Required

Remark

This failure model is valid for Lagrangian solid element orthotropic materials. (See also the DMATORentry.)

FAILSDT Maximum Plastic Strain and Minimum Time-Step Failure Model

Defines the properties of a failure model where total failure occurs when the equivalent plastic strain exceeds the specified value and the element time step falls below the specified limit.

Format and Example

1	2	3	4	5	6	7	8	9	10
FAILSDT	FID	MPS	DT						
FAILSDT	1	.15	1.E-9						

Field	Contents	Type	Default
FID	Unique failure model number	I > 0	Required
MPS	Maximum plastic strain that causes failure on the deviatoric part of the stress tensor	R	Required
DT	Minimum time step that causes total failure	R	Required

Remarks

1. This failure model is valid for Lagrangian solid element materials. (See also the DMAT entry.)
2. The FAILSDT failure model is a two-stage failure. The first stage retains the hydrodynamic properties of the material. The second stage is reached when the global time step falls below the specified value. The element then is removed from the computation.

FFCONTR — Closed Volume Intended for Fluid Filled Containers

Defines the pressure within a closed volume. Intended for the use in (partially) filled containers, where dynamic fluid effects are negligible; e.g. top loading and hot filling.

Format and Example

1	2	3	4	5	6	7	8	9	10
FFCONTR	GID	SID	FVOL	PATM	TEMPTAB	DENSTAB	TACTIVE	PINIT	
FFCONTR	1	2	1.50E-03	0.1E6	10	20			

Field	Contents	Type	Default
GID	Unique FFCONTR number	I > 0	Required
SID	Surface number	I > 0	Required
FVOL	Fluid volume in the container	R ≥ 0.0	Required
PATM	Atmospheric pressure. Used for determination of the constant C for $p \cdot V = C$.	R > 0.0	Required
TEMPTAB	A reference to a TABLED1 ID that specifies how temperature of the container changes in time. So, temperature is a function of time	I > 0	
DENSTAB	Reference of a TABLED1 ID that specifies how density of the container changes with temperature. So, density is a function of temperature.	I > 0	
TACTIVE	Time at which the pressure computation inside the bottle equals the ambient pressure. The volume of the bottle at TACTIV will be used for the finial pressure computation of the gas in the bottle.	R > 0.0	0
PINIT	Initial pressure of the air inside the container.	R > 0.0	PATM

Remarks

1. The gas above the fluid is assumed to be an ideal, iso-thermal gas: $p \cdot V = C$, where C is a constant. If TEMPTAB is set, the temperature is applied to both the fluid as well as the gas. Then, the gas satisfies $p \cdot V/T = C$, where T is the temperature of the fluid. The temperature T has to be given in Kelvin.
2. When DENSTAB is not used, the fluid is assumed to be incompressible.
3. The pressure is based on the uniform pressure gas bag algorithm, where the pressure is uniform in the volume, but variable in time.
4. Output for the fluid-filled container is available through a GBAGOUT definition. The available variables are: PRESSURE, VOLUME, TEMPTURE, VOLGAS, VOLFLUID, GAUGEPRES, and RHOFLUID.
5. The normals of the surface referenced by SID are automatically reversed if required.
6. Modeling guidelines are described in the “Getting Started” section.

7. If DENSTAB is set, then the volume of the fluid changes according to

$$V^{Fluid} = \frac{\rho(T_0)V_0^{Fluid}}{\rho(T)}$$

Here, T_0 and V_0^{Fluid} are initial values for temperature and fluid volume, ρ is the fluid density, and T denotes the current temperature. If TEMPTAB is not set, the DENSTAB entry will not be used.

8. At time = TACTIVE, the gas is assumed to be in contact with the ambient pressure for the last time. This means that at Time = TACTIVE, the pressure in the bottle equals the ambient pressure for the last time. After TACTIVE, the bottle has been closed and there is no longer contact between ambient and gas inside the bottle. Any change in volume of the bottle or temperature or fluid will result in change of pressure of the gas inside the bottle.
9. If PINIT is blank, then PINIT is set equal to the ambient air pressure.

FLOW

Flow Boundary Condition

Defines the properties of a material for the boundaries of a Eulerian mesh.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOW	LID	SID	TYPE1	VALUE1	TYPE2	VALUE2	TYPE3	VALUE3	+
FLOW	120	122	XVEL	100.0					+
+	TYPE4	VALUE4							
+									

Field	Contents		Type	Default
LID	Number of a set of flow boundary conditions		I > 0	Required
SID	Number of a set of segments, specified by CSEG, CFACE, or CFACE1 entries, where the flow boundary is located.		I > 0	Required
TYPEi	The flow boundary property being defined:		C	
	MATERIAL	The material number.		
	XVEL	The material velocity in the x-direction.		
	YVEL	The material velocity in the y-direction.		
	ZVEL	The material velocity in the z-direction.		
	PRESSURE	The pressure of the material at the boundary.		
	DENSITY	The density of the material at inflow.		
	SIE	The specific internal energy at inflow.		
	FLOW	The type of flow boundary required.		
	HYDSTAT	A Hydrostatic pressure profile using a HYDSTAT entry.		
VALUEi	The value for the property defined:		R or C	Required
	For TYPEi set to FLOW, the value is a character entry being either IN, OUT, BOTH, or SYM defining that the flow boundary is defined as an inflow, outflow, or possibly an inflow, outflow, or symmetry boundary. The default is BOTH.			
	VALUEi is required data only if one or more of the TYPEi entries are defined. The TYPEi entries are not required. Thus, a flow boundary by default allows for in- or outflow of the material adjacent to the boundary.			
	For TYPE=HYDSTAT, the value is an integer entry denoting the HYDSTAT entry to be used.			

Remarks

1. LID must be referenced by a TLOAD1entry.
2. Any material properties not specifically defined have the same value as the element with the flow boundary condition.
3. TLOAD entries referencing FLOW entries must have the TID field blank or zero.
4. In the case of material flow into a multi-material Euler mesh, the density and specific energy have to be set. On the other hand, when material flows out of a multi-material Euler mesh, it is assumed that each of the materials present in the outflow Euler element contributes to the out-flow of mass. The materials are transported in proportion to their relative volume fractions.
5. Prescribing both pressure and velocity may lead to the instabilities.
6. For TYPE=HYDSTAT, the pressure is set using HYDSTAT; the velocity equals the element velocity. In case of inflow, the density follows from the hydrostatic pressure by using the EOS.

FLOWC

Cyclic Flow Boundary Condition

Defines the properties of a material for the boundaries of an Eulerian mesh.

Inflow values can be taken from another boundary condition. This allows cyclic or periodic boundary conditions. Likewise, the outflow of material goes into the other boundary condition.

FLOWC* entries have to be defined in pairs. The FID on one entry has to be equal to FID2 of the other entry. For example

```
FLOWC,10,30,20
FLOWC,20,40,10.
```

It is allowed to couple a **FLOWC** with a FLOWCDR or a FLOWCSQ.

For both flow boundaries, mass flow summaries can be created as time history.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWC	FID	SID	FID2						+CONT1
FLOWC	120	122							+CONT1
+CONT1	METHOD	FLOW	ISCALE						
+CONT1	1								

Field	Contents		Type	Default
FID	Unique FLOWC number.		I > 0	Required
SID	Number of a set of segments, specified by CSEG, CFACE, or CFACE1 entries, where the flow boundary is located.		I > 0	Required
FID2	Inflow and Outflow values will be taken from this boundary condition.		I > 0	Required
METHOD	How material properties and pressure are mapped going from one boundary to the other. There are three methods:		I	1
	1	Map both velocity, material flow properties, and pressure loads one-to-one.		
	2	Map velocity, material flow one-to-one. Pressure is given by element pressures		
	3	Inflow is taken as the average of outflow properties. This applies to both velocity and material properties. Pressure is given by element pressures.		

Field	Contents	Type	Default
FLOW	Being either IN, OUT, or BOTH defining that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH.		
ISCALE	Table ID that specifies a time-dependent scale factor by which the amount of inflow is multiplied. This will not conserve total mass. In addition, at the outflow boundary, the work done by pressure forces (the expansion work) is not taken into account. Consequently, when this table ID has been defined, the computation method is an approximation for any value of the scaling factor in the table, including for a value of 1.0.	I > 0	

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWC entries must have the TID field blank or zero.
3. FLOWC can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.
4. FLOWC is only supported by the multi-material Euler solver.
5. The boundary condition FID2 has also to be defined by means of a FLOWCDR or FLOWCSQ. definition. The outflow values of FID2 will be used as the inflow values for FID and vice versa.
6. To enable using the outflow values of FID2 as the inflow values for FID, the boundary faces of FID and FID2 are mapped onto each other. For this mapping, only translations and rotations around coordinate axes are permitted.
7. For fully cyclic boundary conditions, METHOD can be left blank. The default value 1 will be taken.
8. Consider a cubic Euler mesh and that material flows from the left to the right. At the right side, the boundary condition is imposed

   ```
   FLOWC,4,30,,5
   ```

   and at the left side

   ```
   FLOWC,5,40,,4
   ```

   These definitions cause all material that flows out of the right side boundary into the left side boundary. Moreover, the Euler element pressures on the right side are put on the Euler elements of the left boundary condition.

   In practice, it may be useful to skip the coupling between the two boundaries with regard to pressure. This can be done by setting METHOD equal to 2.

   With these definitions, material from boundary 4 still flows into boundary 5 but pressure boundaries are transmitting.

9. ISCALE is useful if several objects have identical outflow that is used as inflow by another object. Then only one object has to be modeled. To account for the other objects when defining inflow, the scale factor can be used. It can also be used to turn off in and outflow. When ISCALE is set either METHOD=2 or METHOD=3 are recommended. METHOD=1 should not be used.
10. It is allowed that the definition of the **FLOWC** entry overlaps with FLOWSQ, FLOW, or FLOWDIR definitions. In that case, the **FLOWC** definition overrules the other ones.

FLOWCDR Cyclic Flow Boundary Condition for All Faces in Certain Direction

Defines the properties of a material for the boundaries of an Eulerian mesh. The boundary consists of all Eulerian faces that are pointing into a specific direction.

Inflow values can be taken from another boundary condition. This allows cyclic or periodic boundary conditions. Likewise, the outflow of material goes into the other boundary condition.

FLOWC* entries have to be defined in pairs. The FID on one entry has to be equal to FID2 of the other entry. For example

```
FLOWCDR,10,MMHYDRO,,10,POSX,2,...
FLOWCDR,20,MMHYDRO,,20,NEGX,1,..
```

It is allowed to couple a **FLOWCDR** with a FLOWC or a FLOWCSQ.

For both flow boundaries, mass flow summaries can be created as time history.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWCDR	FID	ELTYPE	MESH	DIRECTION	FID2				+CONT1
FLOWCDR	120	HYDRO		POSX	140				+CONT1
+CONT1	METHOD	FLOW	ISCALE						
+CONT1	1								

Field	Contents	Type	Default
FID	Unique FLOWCDR number.	I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are: HYDRO, MMHYDRO, and MMSTREN.	C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied.		See Remark 8.
DIRECTION	Allowed values are NEGX, POSX, NEGY, POSY, NEGZ, and POSZ.		
FID2	Inflow and Outflow values will be taken from this boundary condition.		
METHOD	How material properties and pressure are mapped going from one boundary to the other. There are three methods:	I	1
	1 Map both velocity, material flow properties, and pressure loads one-to-one.		
	2 Map velocity, material flow one-to-one. Pressure is given by element pressures.		

Field	Contents	Type	Default
	3 Inflow is taken as the average of outflow properties. This applies to both velocity and material properties. Pressure is given by element pressures.		
FLOW	Being either IN, OUT, or BOTH defining that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH.		
ISCALE	Table ID that specifies a time-dependent scale factor by which the amount of inflow is multiplied. This will not conserve total mass. In addition, at the outflow boundary, the work done by pressure forces (the expansion work) is not taken into account. Consequently, when this table ID has been defined, the computation method is an approximation for any value of the scaling factor in the table, including for a value of 1.0.	I > 0	

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWCDR entries must have the TID field blank or zero.
3. FLOWCDR can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.
4. In the OUT file, the total area of boundary faces is printed.
5. FLOWCDR is only supported by the multi-material Euler solver.
6. The boundary condition FID2 has also to be defined by means of a FLOWC or FLOWCSQ definition. The outflow values of FID2 will be used as the inflow values for FID and vice versa.
7. To enable using the outflow values of FID2 as the inflow values for FID, the boundary faces of FID and FID2 are mapped onto each other. For this mapping, only translations and rotations around coordinate axes are permitted.
8. For fully cyclic boundary conditions, METHOD can be left blank. The default value 1 will be taken.
9. Consider a cubic Euler mesh and that material flows from the left to the right. At the right side, the boundary condition is imposed

   ```
   FLOWCDR,4,MMHYDRO,1,POSX,5
   ```

   and at the left side

   ```
   FLOWCDR,5,MMHYDRO,2,NEGX,4
   ```

   These definitions cause all material that flows out of the right side boundary into the left side boundary. Moreover, the Euler element pressures on the right side are put on the Euler elements of the left boundary condition.

   In practice, it may be useful to skip the coupling between the two boundaries with regard to pressure. This can be done by setting METHOD equal to 2.

With these definitions, material from boundary 4 still flows into boundary 5 but pressure boundaries are transmitting.

10. ISCALE is useful if several objects have identical outflow that is used as inflow by another object. Then only one object has to be modeled. To account for the other objects when defining inflow, the scale factor can be used. It can also be used to turn off in and outflow. When ISCALE is set either METHOD=2 or METHOD=3 are recommended. METHOD=1 should not be used.
11. It is allowed that the definition of the **FLOWCDR** entry overlaps with FLOWSQ, FLOW, or FLOWDIR definitions. In that case, the **FLOWCDR** definition overrules the other ones. It is not allowed that a square of one **FLOWCDR** overlaps a square of another **FLOWCDR** entry.
12. The MESH-ID is only used when multiple Euler domains have been defined. If multiple Euler domains have been defined but if the MESH-ID is blank, all Euler domains will be considered in assigning the boundary condition.

FLOWCSQ Cyclic Flow Boundary Condition using a Square Definition

Defines the properties of a material for the boundaries of an Eulerian mesh. The boundary consists of all Eulerian faces that are inside a specified square.

Inflow values can be taken from another boundary condition. This allows cyclic or periodic boundary conditions. Likewise, the outflow of material goes into the other boundary condition.

FLOWC* entries have to be defined in pairs. The FID on one entry has to be equal to FID2 of the other entry. For example

```
FLOWCSQ,10,MMHYDRO,10,,2
FLOWCSQ,20,MMHYDRO,20,,1
```

It is allowed to couple a **FLOWCSQ** with a FLOWC or a FLOWCDR.

For both flow boundaries, mass flow summaries can be created as time history.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWCSQ	FID	ELTYPE	MESH	FID2					+CONT1
FLOWCSQ	120	HYDRO							+CONT1
+CONT1	XMIN	XMAX	YMIN	YMAX	ZMIN	ZMAX			+CONT2
+CONT1	0.1		0	0	0	0.2			+CONT2
+CONT2	METHOD	FLOW	ISCALE						
+CONT2	1								

Field	Contents	Type	Default
FID	Unique FLOWCSQ number.	I > 0	Required
FID2	Inflow and Outflow values will be taken from this boundary condition.	I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are: HYDRO, MMHYDRO, and MMSTREN.	C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied.		See Remark 8.
XMIN-ZMAX	Defines a square by specifying the ranges of the x,y,z coordinates. For a square in, for example, the x-plane it is required that either XMIN = XMAX or that XMAX is left blank. For at least one coordinate direction the MIN value has to be set	R	See Remark 6.

Field	Contents	Type	Default
METHOD	How material properties and pressure are mapped going from one boundary to the other. There are three methods:	I	1
	1 Map both velocity, material flow properties, and pressure loads one- to-one.		
	2 Map velocity, material flow one-to-one. Pressure is given by element pressures.		
	3 Inflow is taken as the average of outflow properties. This applies to both velocity and material properties. Pressure is given by element pressures.		
FLOW	Being either IN, OUT, or BOTH defining that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH.		
ISCALE	Table ID that specifies a time-dependent scale factor by which the amount of inflow is multiplied. This will not conserve total mass. In addition, at the outflow boundary, the work done by pressure forces (the expansion work) is not taken into account. Consequently, when this table ID has been defined, the computation method is an approximation for any value of the scaling factor in the table, including for a value of 1.0.	I > 0	

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWCSQ entries must have the TID field blank or zero.
3. FLOWCSQ can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.
4. In the OUT file the total area of boundary faces is printed.
5. FLOWCSQ is only supported by the multi material Euler solver.
6. The boundary condition FID2 has also to be defined by means of a FLOWC or FLOWCDR definition. The outflow values of FID2 will be used as the inflow values for FID and vice versa.
7. To enable using the outflow values of FID2 as the inflow values for FID the boundary faces of FID and FID2 are mapped onto each other. For this mapping only translations and rotations around coordinate axes are permitted.
8. For fully cyclic boundary conditions, METHOD can be left blank. The default value 1 will be taken.
9. Consider a cubic Euler mesh and that material flows from the left to the right. At the right side the boundary condition is imposed

   FLOWCSQ,4,MMHYDRO,,5

   and at the left side

```
FLOWCSQ,5,MMHYDRO,,4
```

These definitions cause all material that flows out of the right side boundary into the left side boundary. Moreover, the Euler element pressures on the right side are put on the Euler elements of the left boundary condition.

In practice, it may be useful to skip the coupling between the two boundaries with regard to pressure. This can be done by setting METHOD equal to 2.

With these definitions, material from boundary 4 still flows into boundary 5 but pressure boundaries are transmitting.

10. ISCALE is useful if several objects have identical outflow that is used as inflow by another object. Then only one object has to be modeled. To account for the other objects when defining inflow, the scale factor can be used. It can also be used to turn off in and outflow. When ISCALE is set either METHOD=2 or METHOD=3 are recommended. METHOD=1 should not be used.
11. If neither the MIN or MAX value has been set, the default value is respectively -1E+20 and 1e+20 for the MIN and MAX value. If the MIN value has been set, the default value of the MAX value is the MIN value.
12. It is allowed that the definition of the **FLOWCSQ** entry overlaps with FLOWSQ, FLOW, or FLOWDIR definitions. In that case, the **FLOWCSQ** definition overrules the other ones. It is not allowed that a square of one **FLOWCSQ** overlaps a square of another **FLOWCSQ** entry.
13. The MESH-ID is only used when multiple Euler domains have been defined. If multiple Euler domains have been defined but if the MESH-ID is blank, all Euler domains will be considered in assigning the boundary condition.

FLOWDEF

Default Flow Boundary

Definition of default Eulerian flow boundary condition.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWDEF	FID		TYPEM						+
FLOWDEF	25		HYDRO						+
+	TYPE1	VALUE1	TYPE2	VALUE2	-etc.-				
+	DENSITY	1000.							

Field	Contents		Type	Default
FID	Unique FLOWDEF number		I > 0	Required
TYPEM	HYDRO, STRENGTH, MMHYDRO, or MMSTREN		C	HYDRO
TYPEi	The flow boundary property being defined:		C	
	MATERIAL	The material number.		
	XVEL	The material velocity in the x-direction.		
	YVEL	The material velocity in the y-direction.		
	ZVEL	The material velocity in the z-direction.		
	PRESSURE	The pressure of the material at the boundary.		
	DENSITY	The density of the material at inflow.		
	SIE	The specific internal energy at inflow.		
	FLOW	The type of flow boundary required.		
	HYDSTAT	A Hydrostatic pressure profile using a HYDSTAT entry		
VALUEi	The value for the property defined:		R or C	Required
	For TYPEi set to flow the value is a character entry being either IN, OUT, or BOTH, defining that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH.			
	VALUEi is required data only if one or more of the TYPEi entries are defined. The TYPEi entries are not required. Thus, a flow boundary by default allows for in- or outflow of the material adjacent to the boundary.			
	For TYPE=HYDSTAT, the value is an integer entry denoting the HYDSTAT entry to be used.			

Remark

1. If this entry is not specified, a default wall boundary condition is applied to all Eulerian free faces.
2. For `TYPE=HYDSTAT`, the pressure is set using `HYDSTAT`; the velocity equals the element velocity. In case of inflow, the density follows from the hydrostatic pressure by using the `EOS`.

FLOWDIR

Flow Boundary Condition for all Eulerian Boundary Faces in a Specific Direction

Defines the properties of a material for the boundaries of an Eulerian mesh. The boundary consists of all Eulerian boundary faces that point in a specific direction.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWDIR	FID	ELTYPE	MESH	DIRECTION					+CONT1
FLOWDIR	120	HYDRO		NEGX					+CONT1
+CONT1	TYPE1	VALUE1	TYPE2	VALUE2					
+CONT1	XVEL	10.0							

Field	Contents		Type	Default
FID	Unique FLOWDIR number		I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are: HYDRO, MMHYDRO and MMSTREN.		C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied		I	See Remark 7.
DIRECTION	Allowed values are NEGX, POSX, NEGY, POSY, NEGZ, and POSZ.		C	Required
TYPEi	The flow boundary property being defined.		C	
	MATERIAL	The material number.		
	XVEL	The material velocity in the x-direction.		
	YVEL.	The material velocity in the y-direction		
	ZVEL	The material velocity in the z-direction		
	PRESSURE	The pressure of the material at the boundary.		
	DENSITY	The density of the material at inflow.		
	SIE	The specific internal energy at inflow.		
	FLOW	The type of flow boundary required.		
	HYDSTAT	A Hydrostatic pressure profile using an HYDSTAT entry.		
	MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied	I	See Remark 7.

Field	Contents	Type	Default
VALUEi	The value for the property defined.	R or C	Required
	For TYPEi set to FLOW, the value is a character entry being either IN, OUT, BOTH, or SYM defining that the flow boundary is defined as an inflow, outflow, or possibly an inflow, outflow, or symmetry boundary. The default is BOTH.		
	VALUEi is required data only if one or more of the TYPEi entries are defined. The TYPEi entries are not required. Thus, a flow boundary by default allows for in- or outflow of the material adjacent to the boundary.		
	For TYPE = HYDSTAT, the value is an integer entry denoting the HYDSTAT entry to be used.		

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWDIR entries must have the TID field blank or zero.
3. FLOWDIR can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH, BOX option.
4. In the OUT file, the total area of boundary faces is printed.
5. FLOWDIR is not supported by the single material strength Euler solver.
6. FLOWDIR overrules FLOW definitions, but FLOWSQ overrules FLOWDIR.
7. The MESH-ID is only used when multiple Euler domains have been defined. If multiple euler domains have been defined but if the MESH-ID is blank all Euler domains will be considered in assigning the boundary condition.
8. Any material properties not specifically defined have the same value as the element with the flow boundary condition.
9. TLOAD entries referencing FLOW entries must have the TID field blank or zero.
10. In the case of material flow into a multi-material Euler mesh, the density and specific energy have to be set. On the other hand, when material flows out of a multi-material Euler mesh, it is assumed that each of the materials present in the outflow Euler element contributes to the out flow of mass. The materials are transported in proportion to their relative volume fractions.
11. Prescribing both pressure and velocity may lead to the instabilities.
12. For TYPE = HYDSTAT, the pressure is set using HYDSTAT, the velocity equals the element velocity. In case of inflow the density follows from the hydrostatic pressure by using the equation of state.

FLOWEX

User-defined Flow Boundary

Defines a flow boundary specified by a user subroutine.

Example in FMS Section of the Dytran input stream:

CONNECT SERVICE myflow SCA.MDSolver.Obj.Uds.Dytran.Flow

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWEX	LID	SID	NAME	MULTI	GROUP				
FLOWEX	150	300	PRES1		myflow				

Field	Contents	Type	Default
LID	Number of a set of flow boundary conditions	I > 0	Required
SID	Number of a set of segments, specified by CSEG or CFACE entries, where the flow boundary is located.	I > 0	Required
NAME	Name of the flow boundary (See also Remark 6.)	C	Required
MULTI	Enables inflow of multiple materials. Allowed values are OFF and ON (see Remark 2 and 7).	C	OFF
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. LID must be referenced by a TLOAD1 entry.
2. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual. The usrFlow (cpp) or ext_flow (Fortran) user subroutines are available for single hydrodynamic materials, or materials with strength material problems. The usrFlow3 (cpp) or ext_flow3 (Fortran) user subroutines are available for multi-material problems. MULTI=ON option must be used only for multi-material problems.
3. TLOAD1 entries referencing FLOWEX entries must have the TID field blank or zero.
4. The flow boundary LID is passed to the usrFlow (cpp), usrFlow3(cpp), ext_flow (Fortran) and ext_flow3(Fortran) user subroutines and can be used to identify the boundary.
5. The usrFlow3 (cpp) or ext_flow3 (Fortran) user subroutines allows for the definition of any material to flow into the Eulerian mesh. The outflow can only be of materials present in the mesh.
6. There are two methodologies available to define an inflator model for an eulerian calculation:
   a. As a boundary condition for a subsurface on a coupling surface (see the COUPLE, COUPOR, and INFLATR entries)
   b. As a FLOWEX boundary condition for an Euler face.

The second method can be activated by using a predefined name on the FLOWEX entry. The following name must be used:

7. If MULTI = ON, usrFlow3 (cpp) or ext_flow3 (Fortran) user subroutines has to be supplied. The use of this routine is only supported for multiple material Euler solver.

FLOWSQ

Flow Boundary Condition using a Square Definition

Defines the properties of a material for the boundaries of an Eulerian mesh. The boundary consists of all Eulerian faces that are inside a specified square.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWSQ	FID	ELTYPE	MESH						+CONT1
FLOWSQ	120	HYDRO							+CONT1
+CONT1	XMIN	XMAX	YMIN	YMAX	ZMIN	ZMAX			+CONT2
+CONT1	0.1		0	0.2	0	0.2			+CONT2
+CONT2	TYPE1	VALUE1	TYPE2	VALUE2					
+CONT2	XVEL	10							

Field	Contents		Type	Default
FID	Unique FLOWSQ number		I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are: HYDRO, MMHYDRO, and MMSTREN.		C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied			See Remark 8.
XMIN-ZMAX	Defines a square by specifying the ranges of the x, y, z coordinates. For a square in, for example, the x-plane, it is required that either XMIN = XMAX or that XMAX is left blank.		R	See Remark 6.
	For at least one coordinate direction, the MIN value has to be set			
TYPEi	The flow boundary property being defined.		C	
	MATERIAL	The material number.		
	XVEL	The material velocity in the x-direction.		
	YVEL	The material velocity in the y-direction.		
	ZVEL	The material velocity in the z-direction.		
	PRESSURE	The pressure of the material at the boundary.		
	DENSITY	The density of the material at inflow.		

Field	Contents		Type	Default
	SIE	The specific internal energy at inflow		
	FLOW	The type of flow boundary required.		
	HYDSTAT	A Hydrostatic pressure profile using an HYDSTAT entry.		
VALUEi	The value for the property defined.		R,I or C	Required
	For TYPEi set to FLOW, the value is a character entry being either IN, OUT, BOTH, or SYM defining that the flow boundary is defined as an inflow, outflow, or possibly an inflow, outflow, or symmetry boundary. The default is BOTH.			
	VALUEi is required data only if one or more of the TYPEi entries are defined. The TYPEi entries are not required. Thus, a flow boundary by default allows for in- or outflow of the material adjacent to the boundary.			
	For TYPE = HYDSTAT, the value is an integer entry denoting the HYDSTAT entry to be used.			

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWSQ entries must have the TID field blank or zero.
3. FLOWSQ can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.
4. In the OUT file, the total area of boundary faces is printed.
5. FLOWSQ is not supported by the single material strength Euler solver.
6. If either the MIN or MAX value has been set the default value is respectively -1E+20 and 1e+20 for the MIN and MAX value. If the MIN value has been set the default value of the MAX value is the MIN value.
7. It is allowed that the defined square of a FLOWSQ entry overlaps with FLOW definitions or FLOWDIR definitions. In that case, the FLOWSQ definition overrules the other ones. It is not allowed that a square of one FLOWSQ overlaps a square of another FLOWSQ entry.
8. The MESH-ID is only used when multiple Euler domains have been defined. If multiple euler domains have been defined but if the MESH-ID is blank all Euler domains will be considered in assigning the boundary condition.
9. Any material properties not specifically defined have the same value as the element with the flow boundary condition.
10. In the case of material flow into a multi-material Euler mesh, the density and specific energy have to be set. On the other hand when material flows out of a multi-material Euler mesh it is assumed that each of the materials present in the outflow Euler element contributes to the out flow of mass. The materials are transported in proportion to their relative volume fractions.
11. Prescribing both pressure and velocity may lead to the instabilities.

12. For `TYPE = HYDSTAT`, the pressure is set using `HYDSTAT`, the velocity equals the element velocity. In case of inflow, the density follows from the hydrostatic pressure by using the EOS.

FLOWT

Time-dependent Flow Boundary

Definition of the material properties for the inflow or outflow through the boundary of an Euler mesh. Inflow velocity and material properties can be chosen as time dependent.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWT	FID	SID	TYPE						+
FLOWT	2	122	IN						+
+	VELTYPE	VELOCITY	PRESTYP	PRES					+
+	TABLE	101	TABLE	102					+
+	MID	DENSTYP	DENSITY	SIETYPE	SIE				
+	91	TABLE	104	TABLE	107				

Field	Contents		Type	Default
FID	Unique number of a FLOWT entry		I > 0	Required
SID	Number of a set of segments specified by CSEG, CFACE, or CFACE1 entries where the flow boundary is located.		I > 0	Required
TYPE	IN	Inflow boundary (see Remarks 2.and 3.)	C	Required
		Only inflow is allowed. The inflow velocity and pressure can be optionally specified. If not given, the values in the adjacent Euler element is used. The same holds for the DENSITY and SIE.		
	OUT	Only outflow is allowed. The inflow velocity and pressure can be optionally specified. If not given, the values in the adjacent Euler element are used. The outflow boundary always uses the material mixture present in the adjacent Euler element.		
	BOTH	Material is allowed to flow in or out. In- or outflow is based on the direction of the velocity in the adjacent Euler element. Only pressure can be optionally defined. If not given, the pressure in the adjacent Euler element is taken.		
VELTYPE	Type of velocity definition:		C	Element
	ELEMENT	Value of Euler element.		
	CONSTANT	Value is constant in time.		
	TABLE	Value varies in time.		

Field	Contents		Type	Default
VELOCITY	Value of inflow or outflow velocity. If VELTYPE = TABLE, it refers to a TABLED1 or TABLEEX ID. The velocity direction is normal to the segment. A positive velocity corresponds with inflow.		I or R	See Remark 5.
PRESTYP	Type of pressure definition:		C	See Remark 5.
	ELEMENT	Value of Euler element.		
	CONSTANT	Value is constant in time.		
	TABLE	Value varies in time.		
PRES	Value of inflow or outflow pressure. If PRESTYPE = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	
MID	Material ID of inflowing material. Input is not allowed for TYPE = OUT.		I	
	When MID is specified, it is required to also define density and SIE for the inflowing material.			
DENSTYP	Type of density definition		C	Required when MID is given.
	ELEMENT	Value of Euler element.		
	CONSTANT	Value is constant in time.		
	TABLE	Value varies in time.		
DENSITY	Value of density. If DENSTYP = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	Required when MID is given.
SIETYPE	Type of density definition		C	Required when MID is given.
	ELEMENT	Value of Euler element.		
	CONSTANT	Value is constant in time.		
	TABLE	Value varies in time.		
SIE	Value of specific internal energy. If SIETYPE = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	Required when MID is given.

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD1 entries referencing FLOWT entries must have the TID field blank or zero.
3. Any material properties not specifically defined have the same value as the element with that boundary condition.
4. In the case of material flow into a multi-material Euler mesh, the material number, the density, and specific energy have to be set. On the other hand, when material flows out of a multi-material Euler mesh, it is assumed that each of the materials present in the outflow Euler element contributes to the outflow of mass. The materials are transported in proportion to their relative volume fractions.
5. The boundary condition initiates or determines a wave in compressible material like gas and water. This can be either an outgoing or an ingoing wave. For stability, it is important that the waves created are compatible with the flow type near the boundary. Relevant flow types are subsonic inflow, subsonic outflow, supersonic inflow, and supersonic outflow. For example, for subsonic inflow, prescribing both pressure and velocity would initiate outgoing waves. Outgoing waves for an inflow boundary condition is known to be instable. However, for supersonic inflow, you can specify both pressure and velocity since there are no outgoing waves at a supersonic inflow boundary.
6. When a TABLEEX is referenced, the EXFUNC user subroutine must be created. See TABLEEX for more details.

FLOWTSQ

Time dependent Flow Boundary

Defines the material properties for the in- or outflow of material trough the boundary of an Euler mesh. Inflow velocity and material properties can be chosen time dependent.

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWTSQ	FID	ELTYPE	MESH						+
FLOWTSQ	2	HYDRO							+
+	XMIN	XMAX	YMIN	YMAX	ZMIN	ZMAX			+
+	0.1		0	0.2	0	0.2			+
+	FLOW	VELTYPE	VELOCITY	PRESTYP	PRES				+
+	IN	TABLE	101	TABLE	102				+
+	MID	DENSTYP	DENSITY	SIETYPE	SIE				
+	91	TABLE	104	TABLE	107				

Field	Contents		Type	Default
FID	Unique number of a FLOWT entry.		I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are: HYDRO, MMHYDRO, and MMSTREN.		C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied		I>0	See Remark 8.
XMIN-ZMAX	Defines a square by specifying the ranges of the x, y, z coordinates. For a square in for example the x-plane it is required that either XMIN = XMAX or that XMAX is left blank.		R	See Remark 6.
	For at least one coordinate direction the MIN value has to be set			
TYPE	IN	Inflow boundary (see remark 2 and 3)	C	Required
		Only inflow is allowed. The inflow velocity and pressure can be optionally specified. If not given, the values in the adjacent Euler element will be used. The same holds for the density and SIE.		

Field	Contents		Type	Default
	OUT	Only outflow is allowed. The inflow velocity and pressure can be optionally specified. If not given, the values in the adjacent Euler element will be used. The outflow boundary will always use material mixture as present in the adjacent Euler element.		
	BOTH	Material is allowed to flow in or out. In or outflow is based on the direction of the velocity in the adjacent Euler element. Only pressure can be optionally defined. If not given, the pressure in the adjacent Euler element is taken.		
VELTYPE	Type of velocity definition		C	Element
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		
	TABLE	Value varies in time		
VELOCITY	Value of inflow or outflow velocity. If VELTYPE = TABLE, it refers to a TABLED1 or TABLEEX ID. The velocity direction is normal to the segment. A positive velocity corresponds with inflow.		I or R	See Remark 7.
PRESTYP	Type of pressure definition		C	See Remark 7.
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		
	TABLE	Value varies in time		
PRES	Value of inflow or outflow pressure. If PRESTYPE = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	
MID	Material ID of inflowing material. Input is not allowed for TYPE = OUT		I	
	When MID is specified, it is required to also define density and size for the inflowing material			
DENSTYP	Type of density definition		C	Required when MID is given
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		
	TABLE	Value varies in time		
DENSITY	Value of density. If DENSTYP = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	Required when MID is given

Field	Contents		Type	Default
SIETYPE	Type of density definition		C	Required when MID is given
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		
	TABLE	Value varies in time		
SIE	Value of specific internal energy. If SIETYPE = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	Required when MID is given

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWTSQ entries must have the TID field blank or zero.
3. FLOWTSQ can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.
4. In the OUT file, the total area of boundary faces is printed.
5. FLOWTSQ is not supported by the single material strength Euler solver.
6. If neither the MIN nor MAX value has been set the default value is respectively -1E+20 and 1e+20 for the MIN and MAX value. If the MIN value has been set the default value of the MAX value is the MIN value.
7. It is allowed that the defined square of a FLOWTSQ entry overlaps with FLOW definitions or FLOWDIR definitions. In that case, the FLOWTSQ definition overrules the other ones. It is not allowed that a square of one FLOWTSQ overlaps a square of another FLOWSQ entry.
8. The MESH-ID is only used when multiple Euler domains have been defined. If multiple euler domains have been defined but if the MESH-ID is blank, all Euler domains are considered in assigning the boundary condition.
9. Any material properties not specifically defined have the same value as the element that with the boundary conditions.
10. In the case of material flow into a multi-material Euler mesh, the material number, the density and specific energy have to be set. On the other hand when material flows out of a multi-material Euler mesh it is assumed that each of the materials present in the outflow Euler element contributes to the out flow of mass. The materials are transported in proportion to their relative volume fractions
11. The boundary condition initiates/determines a wave in compressible material like gas and water. This can be either an outgoing or an ingoing wave. For stability it is important that the waves created are compatible with the flow type near the boundary. Relevant flow types are subsonic inflow, subsonic outflow, supersonic inflow and supersonic outflow. For example for subsonic inflow prescribing both pressure and velocity would initiate outgoing waves. Outgoing waves for an inflow boundary condition is known to be instable. However, for supersonic inflow one can specify both pressure and velocity since there are no outgoing waves at a supersonic inflow boundary.

12. When a TABLEEX is referenced, the user-subroutine EXFUNC must be created. See TABLEEX for more details.

FLOWXDR

Eulerian Boundary Faces in Specific Direction

Defines a flow boundary specified by user subroutine. The boundary consists of all Eulerian boundary faces that point in a specific direction.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE myflow SCA.MDSolver.Obj.Uds.Dytran.Flow
```

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWXDR	FID	ELTYPE	MESH	NAME	MULTI	DIRECTION	GROUP		+
FLOWXDR	1	MMHYDRO		INFLOW	OFF	NEGZ	myflow		

Field	Contents	Type	Default
FID	Unique FLOWXDR number of a entry.	I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are HYRDRO, MMHRDRO, and MMSTREN.	C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied	I	See Remark 7.
NAME	Enables inflow of multiple materials (see Remark 9).	C	Required
MULTI	Enables inflow of multiple materials. Allowed values are OFF and ON (see Remark 10).	C	OFF
DIRECTION	Allowed values are NEGX, POSX, NEGY, POSY, MEGZ, and POSZ.	C	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWXDR entries must have the TID field blank or zero.
3. FLOWXDR can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.
4. In the OUT file, the total area of boundary faces is printed.
5. FLOWXDR is not supported by the single material strength Euler solver.
6. FLOWXDR overrules FLOW and FLOWEX definitions, but FLOWSQ and FLOSXSQ overrule FLOWDIR and FLOWXDR.
7. The MESH-ID is only used when multiple Euler domains have been defined. If multiple Euler domains have been defined but the MESH-ID is blank, all Euler domains will be considered in assigning the boundary condition.

8. FID is passed to the usrFlow (cpp), usrFlow3(cpp), ext_flow (Fortran) and ext_flow3(Fortran) user subroutines and can be used to identify the boundary.
9. The usrFlow3(cpp) and ext_flow3(Fortran) user subroutines allow for the definition of any material to flow into the Eulerian mesh. The outflow can only be of materials present in the mesh. The use of this routine is only supported for the multiple material Euler solver.
10. If MULTI = ON, usrFlow3(cpp) and ext_flow3(Fortran) user subroutines have to be supplied. The use of this routine is only supported for the multiple material Euler solver.
11. There are two methodologies available to define an inflator model for an Eulerian calculation:
    a. As a boundary condition for a subsurface on a coupling surface (see the COUPLE, COUPOR, and INFLATR entries).
    b. As a FLOWEX, FLOWXDR, or FLOWSXQ boundary condition for an Euler face.

FLOWXSQ Eulerian Boundary Faces Inside Specified Square

Defines a flow boundary specified by user subroutine. The boundary consists of all Eulerian faces that are inside a specified square.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE myflow SCA.MDSolver.Obj.Uds.Dytran.Flow
```

Format and Example

1	2	3	4	5	6	7	8	9	10
FLOWXSQ	FID	ELTYPE	MESH	NAME	MULTI	GROUP			+CONT1
FLOWXSQ	1	MMHYDRO		INFLOW	OFF	myflow			+CONT1
+CONT1	XMIN	XMAX	YMIN	YMAX	ZMIN	ZMAX			
+CONT1	0.25	0.75	0.25	0.75	0				

Field	Contents	Type	Default
FID	Unique FLOWXSQ number of a entry.	I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are HYRDRO, MMHRDRO, and MMSTREN.	C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied	I	See Remark 7.
NAME	Enables inflow of multiple materials (see Remark 9).	C	Required
MULTI	Enables inflow of multiple materials. Allowed values are OFF and ON (see Remark 11).	C	OFF
DIRECTION	Allowed values are NEGX, POSX, NEGY, POSY, MEGZ, and POSZ.	C	Required
XMIN-ZMAX	Defines a square by specifying the ranges of the x, y, z coordinates. For a square in, for example, the x-plane, it is required that either XMIN = XMAX or that XMAX is left blank. For at least one coordinate direction, the MIN value has to be set.	R	See Remark 6.
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. FID must be referenced by a TLOAD1 entry.
2. TLOAD entries referencing FLOWXSQ entries must have the TID field blank or zero.
3. FLOWXSQ can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.

4. In the OUT file, the total area of boundary faces is printed.
5. FLOWXSQ is not supported by the single material strength Euler solver.
6. If neither the MIN or MAX value has been set, the default value is -1E+20 and 1e+20 for the MIN and MAX value, respectively. If the MIN value has been set, the default value of the MAX value is the MIN value.
7. It is allowed that the defined square of a FLOWXSQ entry overlaps with FLOW, FLOWEX, or FLOWDIR definitions. In that case, the FLOWXSQ definition overrules the other definitions. It is not allowed that a square of one FLOWXSQ overlaps a square of another FLOWXSQ entry.
8. The MESH-ID is only used when multiple Euler domains have been defined. If multiple Euler domains have been defined but the MESH-ID is blank, all Euler domains will be considered in assigning the boundary condition.
9. FID is passed to the usrFlow (cpp), usrFlow3(cpp), ext_flow (Fortran) and ext_flow3(Fortran) user subroutines and can be used to identify the boundary.
10. The usrFlow3(cpp) and ext_flow3(Fortran) user subroutines allows for the definition of any material to flow into the Eulerian mesh. The outflow can only be of materials present in the mesh. The use of this routine is only supported for the multiple material Euler solver.
11. If MULTI = ON, usrFlow3(cpp) and ext_flow3(Fortran) user subroutines have to be supplied. The use of this routine is only supported for the multiple material Euler solver.
12. There are two methodologies available to define an inflator model for an Eulerian calculation:
    a. As a boundary condition for a subsurface on a coupling surface (see the COUPLE, COUPOR, and INFLATR entries).
    b. As a FLOWEX, FLOWXDR, or FLOWSXQ boundary condition for an Euler face.

FOAM1

Foam Material Properties

Defines the properties of an isotropic, crushable material where Poisson's ratio is effectively zero.

Format and Example

1	2	3	4	5	6	7	8	9	10
FOAM1	MID	RHO	G	K	TABLE	TYPE			+
FOAM1	3	0.01		3.	111	CRUSH			+
+			BULKTYP	BULKQ	BULKL				
+			DYNA	1.4	0.05				

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
G	Shear modulus		R > 0.0	See Remark 3.
K	Bulk modulus		R > 0.0	See Remark 3.
TABLE	Number of a TABLED1 entry defining the variation of stress (y-value) with crush factor or true strain (x- value).		I > 0	Required
TYPE	The type of data defined as the x-value in the table:		C	CRUSH
	CRUSH	Crush factor (1, relative volume)		
	STRAIN	True strain. See also Remark 4.		
BULKTYP	Bulk-viscosity model		C	DYNA
	DYNA	Standard DYNA3D model		
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0	1.0
BULKL	Linear bulk-viscosity coefficient		R ≥ 0	0.0

Remarks

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values are used.
2. The continuation line with bulk-viscosity data can be omitted.
3. Poisson's ratio for this model is effectively zero. Therefore, only one other elastic constant can be defined which can be G, the shear modulus, or K, the bulk modulus.

4. For this model, the stress-strain curve is independent of the experimental test performed to obtain the material data (uniaxial, shear, or volumetric). The most common test is the uniaxial compression test where the stress-strain characteristic can either be defined in terms of the amount of crush, which is minus the engineering strain, or in terms of the true strain. Since Poisson's ratio is effectively zero the amount of crush is defined as $\left(1-\frac{V}{V_0}\right)$, with $\frac{V}{V_0}$ as the relative volume, and the true strain is defined as

   $\int_{t_0}^{t} \frac{dV}{V}$ or $In\left(\frac{V}{V_0}\right)$.

   The crush factor must be between 0 and 1. The true strain must always be negative and the stress positive (absolute value).

5. The yield surface in three-dimensional space is a sphere in principal stresses, and is defined by

   $$\tau_{11}^2 + \tau_{22}^2 + \tau_{33}^2 = R_s^2$$

   where the radius of the sphere R_s depends on the strains as follows:

   $$R_s = f(R_e)$$

   with

   $$\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2 = R_e^2$$

   and f is the function defined by the stress-strain table.

6. This material can only be used with Lagrangian solid elements.

FOAM2

Foam Material Properties

Defines the properties of an isotropic, elastic foam material with user-specified hysteresis response for unloading, with strain rate dependency, and where Poisson's ratio is effectively zero.

Format and Example

1	2	3	4	5	6	7	8	9	10
FOAM2	MID	RHO	G	K	TABLE	TYPE	VALUE	CUTOFF	+
FOAM2	3	0.01		3.	111	CRUSH	-100.	SFRAC	+
+	TABY	ALPHA	UNLOAD						+
+	112	0.4	LINEAR						+
+			BULKTYP	BULKQ	BULKL				
+			DYNA	1.4	0.05				

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
G	Shear modulus		R > 0.0	See Remark 3.
K	Bulk modulus		R > 0.0	See Remark 3.
TABLE	Number of a TABLED1 entry defining the variation of stress (y-value) with crush factor or true strain (x- value)		I > 0	Required
TYPE	The type of data defined as the x-value in the table:		C	CRUSH
	CRUSH	Crush factor (=1–relative volume)		
	STRAIN	True strain. See also Remark 4.		
VALUE	The value for cut-off stress		R ≤ 0.0	-0.1 * Young's modulus.
CUTOFF	Cut-off stress:		C	SMIN
	SFRAC	Stress for tensile failure		
	SMIN	Minimum stress		
TABY	Number of a TABLED1 entry giving the variation of the scale factor for the stress (y-value) with the strain rate (x-value).		I > 0	See Remark 7.
ALPHA	Energy dissipation factor		0.0≤R≤1.0	Required
UNLOAD	Unloading option:		C	QDRATIC
	EXPTIAL	Unloading via exponential curve		

Field	Contents		Type	Default
	LINEAR	Unloading via piecewise linear curve		
	QDRATIC	Unloading via quadratic curve		
BULKTYP	Bulk-viscosity model:		C	DYNA
	DYNA	Standard DYNA3D model.		
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0	1.0
BULKL	Linear bulk-viscosity coefficient		R ≥ 0	0.0

Remarks

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values are used.
2. The continuation line with bulk-viscosity data can be omitted.
3. Poisson's ratio for this model is effectively zero. Therefore, only one other elastic constant can be defined which can be G, the shear modulus, or K, the bulk modulus.
4. For this model, the stress-strain curve is independent of the experimental test performed to obtain the material data (uniaxial, shear, or volumetric). The most common test is the uniaxial compression test where the stress-strain characteristic can either be defined in terms of the amount of crush, which is minus the engineering strain, or in terms of the true strain. Since Poisson's ratio is effectively zero, the amount of crush is defined as $\left(1-\frac{V}{V_0}\right)$, with $\frac{V}{V_0}$ as the relative volume, and the true strain is defined as

   $$\int_{t_0}^{} \frac{dV}{V} \text{ or } In\left(\frac{V}{V_0}\right).$$

   The crush factor must be between 0 and 1. The true strain must always be negative and the stress positive (absolute value).
5. The yield surface in three-dimensional space is a sphere in principal stresses, and is defined by

   $$\tau_{11}^2 + \tau_{22}^2 + \tau_{33}^2 = R_s^2$$

   where the radius of the sphere R_s depends on the strains and strain rates as follows:

   $$R_s = f_1(R_e) f_2(R_r)$$

   with

   $$\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2 = R_e^2$$

   and

   $$\dot{\varepsilon}_{11}^2 + \dot{\varepsilon}_{22}^2 + \dot{\varepsilon}_{33}^2 = R_r^2$$

and f_1 is the function supplied in the stress-strain table and f_2 (if defined) is the function supplied in the factor-strain rate table.

6. A minimum (SMIN) or failure (SFRAC) tensile stress can be defined. In the first case this corresponds to a tensile cut-off where the stress cannot fall below the minimum value. In the second case, if the stress falls below the failure stress the element fails and cannot carry tensile loading for the remainder of the analysis. Thus the stress can never become negative again.
7. If TABY is blank, the stress does not vary with strain rate. If TABY has a value, then it references to a TABLED1 entry, which gives the variation with strain rate of the scale factor applied to the stress.
8. The unloading behavior is piecewise linear (LINEAR), quadratic (QDRATIC) or exponential (EXPTIAL). The unloading curve is constructed such that the ratio of the dissipated energy (area between compressive loading and unloading curve) to total energy (area under the loading curve) is equal to the energy dissipation factor alpha. In the case of piecewise linear unloading, Dytran constructs an unloading curve whose segments are parallel to the supplied compression table, except for the first and last segments, which pass respectively through the origin and the point P on the compression curve where the unloading starts. In the case of quadratic unloading, Dytran constructs a quadratic curve starting in the origin and ending in point P. If the quadratic unloading curve falls below the strain axis, then the unloading stress is set to zero. In the case of exponential unloading, the unloading curve is constructed in a similarly to quadratic unloading except for the shape of the curve, which is created from an exponential function instead of a quadratic polynomial.
9. This material can only be used with Lagrangian solid elements.
10. The behavior of this material is described in Materials.

FORCE

Concentrated Load or Velocity

This entry is used in conjunction with a TLOADn entry and defines the location where the load or enforced motion acts, the direction in which it acts, and the scale factor.

Format and Example

1	2	3	4	5	6	7	8	9	10
FORCE	LID	G	CID	SCALE	N1	N2	N3		
FORCE	2	5	1	2.9		1.0			

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid-point number or rigid body where the load is applied	See Remark 4.	Required
CID	Number of a CORDxxx entry	I ≥ 0	0
SCALE	Scale factor for the load	R	1.0
N1, N2, N3	Components of a vector giving the load direction. At least one must be nonzero.	R	See Remark 6.

Remarks

1. At time t, the load $F(t)$ is given by $\overline{F}(t) = SCALE \cdot \overline{N} \cdot T(t)$ where $SCALE$ is the factor; $\overline{N}$ is the vector given by N_1, N_2, and N_3; and $T(t)$ is the value at t interpolated from the table referenced on the TLOADn entry.
2. Concentrated loads can also be defined on the DAREA entry.
3. LID must be referenced by a TLOADn entry.
4. If G references a MATRIG, an RBE2-FULLRIG, or a RIGID surface, the load is applied to the center of the rigid body. If G references a MATRIG, G must be MR<id>, where id is the MATRIG number. If G references an RBE2-FULLRIG, G must be FR<id>, where id is the RBE2 number. If G references a RIGID surface, G is the RIGID surface number.
5. If CID is specified, velocity prescriptions are processed in the local coordinate system referenced by CID. Only velocity prescriptions can be defined in the local coordinate system.
6. If a component field N1, N2, and/or N3 is left blank,

   Force prescription: The component of the force is equal to zero.

   Velocity prescription: The component of the velocity is not restrained.

7. If the TYPE field on the TLOADn entry is 0, it defines a force applied to a grid point. If the TYPE field is 2, it defines an enforced motion on the grid point. If the TYPE field is set to 12, it defines an enforced motion applied to the center of a rigid body, and if the TYPE field is 13, it defines a force applied to the center of a rigid body.

FORCE1

Follower Force, Form 1

This entry is used in conjunction with a TLOADn entry and defines a follower force with direction that is determined by two grid points. FORCE1 applies to any type of grid point.

Format and Example

1	2	3	4	5	6	7	8	9	10
FORCE1	LID	G	SCALE	G1	G2				
ENTRY	2	5	2.9	16	13				

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid point number where the load is applied	I > 0	Required
SCALE	Scale factor for the load	R	1.0
G1, G2	Grid point numbers. The direction of the load is a vector from G1 to G2. G1 must not be equal to G2.	I > 0	Required

Remarks

1. At time t, the load $F(t)$ is given by:

   $$\bar{F}(t) = SCALE \cdot \bar{N} \cdot T(t)$$

   where $SCALE$ is the scale factor, $\bar{N}$ is the unit vector in the direction from G1 to G2, and $T(t)$ is the value at time t interpolated from the table referenced in the TLOAD entry.
2. LID must be referenced by a TLOAD entry.
3. The FORCE1 entry defines a follower force in that the direction of the force changes as the grid points G1 and G2 move during the analysis.

FORCE2

Follower Force, Form 2

This entry is used in conjunction with a TLOADn entry and defines a follower force with a direction that is determined by four grid points. FORCE2 can be applied to any type of grid point.

Format and Example

1	2	3	4	5	6	7	8	9	10
FORCE2	LID	G	SCALE	G1	G2	G3	G4		
FORCE2	2	5	2.9	16	13	17	18		

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid point number where the load is applied	I > 0	Required
SCALE	Scale factor for the load	R	1.0
G1-G4	Grid point numbers. The load direction is determined by a vector product of the vectors from G1 to G2 and G3 to G4. (G1 must not be the same as G2, and G3 must not be the same as G4.)	I > 0	Required

Remarks

1. At time *t*, the load $\overline{F}(t)$ is given by $\overline{F}(t) = SCALE \cdot \overline{N} \cdot T(t)$ where $SCALE$ is the scale factor, $\overline{N}$ is the vector product of the vectors from G1 to G2 and G3 to G4 respectively, and $T(t)$ is the value at time t interpolated from the table referenced by the TLOADn entry.
2. LID must be referenced by a TLOADn entry.
3. The FORCE2 entry defines a follower force in that the direction of the force changes as the grid points G1, G2, G3, and G4 move during the analysis.

FORCE3

Grid Point Velocity Definition

Defines the velocity of a grid point in a local coordinate system or in a cascade of two local coordinate systems.

Format and Example

1	2	3	4	5	6	7	8	9	10
FORCE3	LID	G	CID1	SCALE1	N1	N2	N3		+
FORCE3	77	2		10.	1.	2.5			+
+	CID2	SCALE2	M1	M2	M3				
+									

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid point number	I > 0	Required
CID1	Number of a coordinate system in which N1, N2, and N3 are defined	I ≥ 0	0
SCALE1	Scale factor for the load	R	1.0
N1, N2, N3	Components of a vector giving load direction	R	See Remark 5.
CID2	Number of a coordinate system with respect to which coordinate system CID1 moves with an enforced motion equal to $\overline{M}*SCALE2*F(t)$.	I ≥ 0	0
SCALE2	Scale factor for the enforced rigid-body motion of CID1	R	1.0
M1, M2, M3	Components of a vector giving the enforced motion direction	R	See Remark 5.

Remarks

1. SCALE2 defines the enforced rigid-body motion of the coordinate system referenced by CID1 with respect to the coordinate system referenced by CID2.
2. This boundary condition can be used only to define the enforced velocities of grid points. Thus, the TYPE field in the TLOAD1 or TLOAD2 entry should be set to 2.
3. LID is referenced by a TLOAD entry.
4. If CIDx is specified, the velocity components are defined in the local coordinate directions, for example, if a cylindrical system is referenced, the velocity components define a radial, angular, and axial velocity.

5. If a component field N1, N2, N3, M1, M2, and/or M3 is left blank, that component of the velocity is not restrained.
6. The FORCE3 entry is valid for both Lagrangian and Eulerian grid points.

FORCEEX

User-defined Enforced Motion at Grid Points

Defines enforced motion at grid points specified by a user subroutine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE vel SCA.MDSolver.Obj.Uds.Dytran.Loads
```

Format and Example

1	2	3	4	5	6	7	8	9	10
FORCEEX	LID	NAME	GROUP						+
FORCEEX	120	VEL7	vel						+
+	G1	G2	G3	G4	THRU	G5	-etc.-		
+	100	319	728	429	THRU	457			

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
NAME	Constraint name passed to the user subroutine	C	Required
Gi	Numbers of the grid points that are constrained. If the word THRU appears between two numbers, all the numbers in the range are included in the list. BY indicates the increment to be used within this range.	I > 0	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. LID must be referenced by a TLOAD1 entry.
2. FORCEEX can only be used to specify enforced velocities for grid points. The TYPE field on the TLOAD1 entry must be set to two. The TID on the TLOAD1 entry must be set to zero or blank (no time variation).
3. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual.
4. LID is passed to the subroutine and can be used to identify the constraint.
5. A THRU specification, including the start and finish points in the range, must be on one line.
6. If the THRU specification is used, all the points in the sequence do not have to exist. Those that do not exist are ignored. The first point in the THRU specification must be a valid grid point. BY can be used to exclude grid points.
7. None of the fields in the list of grid points can be blank or zero, since this designation marks the end of the list.
8. Any number of continuation lines can be used to define the list of grid points.

GBAG

Gas Bag Pressure Definition

Defines the Pressure within an Enclosed Volume.

Format and Example

1	2	3	4	5	6	7	8	9	10
GBAG	GID	SID	TRIGGER	TRIGGERV	PORID	INFID	HTRID	INTID	+
GBAG	101	37	TIME	0.0					+
+	CDEX	CDEXV	AEX	AEXV	CDLEAK	CDLEAKV	ALEAK	ALEAKV	+
+	TABLE	201	TABLE	202	TABLE	203	TABLE	204	+
+	FLGAS	FLGASV	TGAS	TGASV	VOLPOR	VOLPORV			+
+	TABLE	205	TABLE	206	TABLE	209			+
+	CPGAS	CPGASV	RGAS	PENV	PEX	REVERSE	CHECK	PINIT	+
+	CONSTANT	1001.							+
+	TINIT	TENV							+
+	293.								+
+	CONVEC	CONVECV	ACONVEC	ACONVECV					+
+									+
+	RADIAT	RADIATV	ARADIAT	ARADIATV	SBOLTZ				
+									

Field	Contents		Type	Default
GID	Unique gas bag number		I > 0	Required
SID	Number of a SURFACE entry defining the geometry of the gas bag		I > 0	Required
TRIGGER	The time-dependent parameters are offset in time.		C	TIME
	TIME	The offset is defined at TRIGGERV.		
TRIGGERV	The value of the offset in time		R	Required
PORID	Number of a set of GBAGPOR entries, that defines the porosity (permeability) and holes for the gas-bag surface and/or subsurfaces.		I > 0	No porosity
INFID	Number of a set of GBAGINFL entries, that defines the one or more inflators on subsurface(s) of the GBAG surface.		I > 0	No inflators

Field	Contents		Type	Default
HTRID	Number of a set of GBAGHTR entries, that defines the heat transfer definitions for the gas-bag surface and/or subsurfaces.		I > 0	No heat transfer
INTID	ID of an INITGAS entry specifying the initial gas composition for this GBAG		I > 0	No initial gas composition
CDEX	The variation of the discharge coefficient for the exhaust openings:		C	CONSTANT
	CONSTANT	The discharge coefficient is constant and is specified in CDEXV.		
	TABLE	The discharge coefficient varies with pressure. CDEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with the pressure (x-value).		
	TIME	The area varies with time. CDEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
CDEXV	The discharge coefficient or the number of a TABLED1 or TABLEEX entry, depending on the value of CDEX. Discharge coefficients must be between zero and one.		R or I > 0	1.0
AEX	The variation of the total area of the exhaust openings.		C	CONSTANT
	CONSTANT	The area is constant and is specified in AEXV.		
	TABLE	The area varies with pressure. AEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value).		
	TIME	The area varies with time. AEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x- value). The table is offset by the time specified on the TRIGGERV entry.		
AEXV	The total area of the exhaust openings or the number of a TABLED1 or TABLEEX entry, depending on the value of AEX.		R or I > 0	0.0
CDLEAK	The variation of the discharge coefficient for the permeability of the gas bag fabric.		C	CONSTANT

Field	Contents		Type	Default
	CONSTANT	The discharge coefficient is constant and is specified in CDLEAKV.		
	TABLE	The discharge coefficient varies with pressure. CDLEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of discharge coefficient (y-value) with the pressure (x-value). The discharge coefficient must be between zero and one.		
	TIME	The discharge coefficient varies with time. CDLEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
CDLEAKV	The discharge coefficient or the number of a TABLED1 or EXFUNC entry, depending on the value of CDLEAK.		R or I > 0	1.0
ALEAK	The variation of the total leak area.		C	CONSTANT
	CONSTANT	The area is constant and is specified in ALEAKV.		
	TABLE	The area varies with pressure. ALEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value).		
	TIME	The area varies with time. ALEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
ALEAKV	The total leak area or the number of a TABLED1 or TABLEEX entry, depending on the value of AEX.		R or I > 0	0.0
FLGAS	The variation of the total mass flux of the inflowing gas. The mass flux is in mass-per-unit time.		C	CONSTANT
	CONSTANT	The mass flux is constant and specified in FLGASV. Flow STARTS at the time specified on the TRIGGERV entry.		
	TABLE	The mass flux varies with time. FLGASV is the number of a TABLED1 or TABLEEX entry giving the variation of the mass flux (y- value) with time (x- value). The table is offset by the time specified on TRIGGERV entry.		

Field	Contents		Type	Default
FLGASV	The mass flux or the number of a TABLED1 or TABLEEX entry, depending on the value of FLGAS.		R or I > 0	Required
TGAS	The variation of the temperature of the inflowing gas.		C	CONSTANT
	CONSTANT	The temperature is constant and specified in TGASV.		
	TABLE	The temperature varies with time. TGASV is the number of a TABLED1 or TABLEEX entry giving the variation of the temperature (y-value) with the time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
TGASV	The temperature of the inflowing gas or the number of a TABLED1 or TABLEEX entry depending on the value of TGAS.		R or I > 0	Required
VOLPOR	User-defined volumetric flow rate volume-per-unit time. See Remark 5..		C	CONSTANT
	CONSTANT	The outflow rate is constant and specified in VOLPORV.		
	TABLE	The outflow rate varies with pressure. VOLPORV is the number of a TABLED1 or TABLEEX entry giving the variation of the outflow rate (y-value) with the pressure (x-value).		
	TIME	The outflow rate varies with time. VOLPORV is the number of a TABLED1 or TABLEEX entry giving the variation of the outflow rate (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
VOLPORV	The flow rate or the number of a TABLED1 or TABLEEX entry, depending on the value of VOLPOR.		R ≥ 0.0 or I > 0	0.0
CPGAS	The variation of the specific heat constant at constant pressure.		C	CONSTANT
	CONSTANT	The specific heat is constant and specified in CPGASV.		
CPGASV	The specific heat of the gas		R	Required
RGAS	Gas constant of the inflowing gas.		R	Required
PENV	Environmental pressure surrounding the gas bag.		R	Required
PEX	There is only outflow from the gas bag if the pressure in the gas bag is greater than PEX.		R	PENV

Field	Contents		Type	Default
REVERSE	Normal auto-reverse switch.		C	ON
	ON	The normals of the SURFACE are automatically reversed if necessary so that they point in the same direction and provide a positive volume.		
	OFF	The normals are not automatically reversed.		
CHECK	Normal checking switch:		C	ON
	ON	The normals of the SURFACE are checked to see if they all point in the same direction and provide a positive volume.		
	OFF	The normals are not checked.		
	If REVERSE is set to ON, CHECK is automatically set to ON.			
PINIT	Initial pressure inside the gas bag.		R	PENV
TINIT	Initial temperature inside the gas bag.		R	Required.
	See Remark 4.			
TENV	Environmental Temperature. The value is required when heat transfer is used.		R > 0.0	Required. See Remark 6.
CONVEC	The variation of the heat transfer coefficient for convection heat transfer.		C	CONSTANT
	CONSTANT	The heat transfer coefficient is constant and specified in CONVECV.		
	TABLE	The heat transfer coefficient varies with time. VONVECV is the number of a TABLED1 or TABLEEX entry giving the variation of the heat transfer coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
CONVECV	The heat transfer coefficient or the number of a TABLED1 or TABLEEX entry, depending on value of CONVEC.		R or I > 0	0.0
ACONVEC	The variation of the total surface area to be used in the convective heat transfer equations. The area is calculated by multiplying the total area of the GBAG surface with the value of this coefficient.			
	CONSTANT	The area coefficient is constant and specified in ACONVECV.		

Field	Contents		Type	Default
	TABLE	The area coefficient varies with time. ACONVECV is the number of a TABLED1 or TABLEEX entry giving the variation of the heat transfer coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
ACONVECV	The area coefficient of the number of a TABLED1 or TABLEEX entry, depending on value of AVONCEC.		R or I > 0	1.0
RADIAT	The variation of the gas emissivity coefficient for radiation heat transfer.		C	CONSTANT
	CONSTANT	The gas emissivity coefficient is constant and specified in RADIATV.		
	TABLE	The gas emissivity coefficient varies with time. RADIATV is the number of a TABLED1 or TABLEEX entry giving the variation of the gas emissivity coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
RADIATV	The gas emissivity coefficient or the number of a TABLED1 or TABLEEX entry, depending on value of RADIAT.		R or I > 0	0.0
ARADIAT	The variation of the total surface area to be used in the radiation heat transfer equations. The area is calculated by multiplying the total area of the GBAG surface with the value of this coefficient.		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified in ARADIATV.		
	TABLE	The area coefficient varies with time. ARADIATV is the number of a TABLED1 or TABLEEX entry giving the variation of the heat transfer coefficient (y-value) with the (x-value). The table is offset by the time specified on the TRIGGERV entry.		
ARADIATV	The area coefficient or the number of a TABLED1 or TABLEEX entry, depending on value of ARADIAT.		R or I > 0	1.0
SBOLTZ	Stefan-Boltzmann constant		R	0.0

Remarks

1. The SURFACE entry referenced by the SID field must form a closed volume.
2. The pressure in the gas bag is applied to all the faces of the SURFACE.

3. TABLEEX is valid also in all entries where TABLED1 is used.
4. TINIT is the temperature of the inflowing gas at $time = 0$. At $time = 0$, the mass of the gas inside the gas bag is calculated as

$$m = \frac{p_{init}V}{RT_{init}}$$

where, p_{init} the initial pressure, V the volume, R the gas constant, and T_{init} the initial gas temperature.

5. The flow through exhaust openings, leakage areas and user-specified outflow rate is accumulated. The volumetric porosity contributes to the outflow of gas as

$$\dot{m}_{out} = \rho * Q = \frac{p}{R * T} * Q$$

where

Q	=	volumetric flow rate
ρ	=	density inside the bag
p	=	pressure inside the bag
R	=	gas constant
T	=	temperature inside the bag
$\dot{m}_{out}$	=	mass outflow rate

The value of Q can be specified as a constant, as a function of the pressure difference, or as a function of time. Negative values for the volumetric flow rate are not allowed, since this would mean inflow of outside air.

6. The heat-transfer rates are given by the following equations:

$$q_{conv} = hA_c(T - T_{env}) \qquad \text{Convection}$$

$$q_{rad} = eA_r(T^4 - T_{env}^4) \qquad \text{Radiation}$$

where h is the convection heat-transfer coefficient (CONVEC, CONVECV), e the gas emissivity coefficient (RADIAT, RADIATV), A_c the air bag surface area for convective heat transfer, A_r the air bag surface area for radiation, and T_{env} the environmental temperature.

GBAGC

Gas Bag Connection

Connection between two gas bags.

Format and Example

1	2	3	4	5	6	7	8	9	10
GBAGC	ID	GID1	GID2	TRIGGER	TRIGGERV	PRESTOL			+
GBAGC	100	11	12	TIME	0.0	0.0			+
+	CD12	CD12V	A12	A12V	CD21	CD21V	A21	A21V	
+	CONSTANT	0.8	CONSTANT	3.0	TABLE	12	TABLE	13	

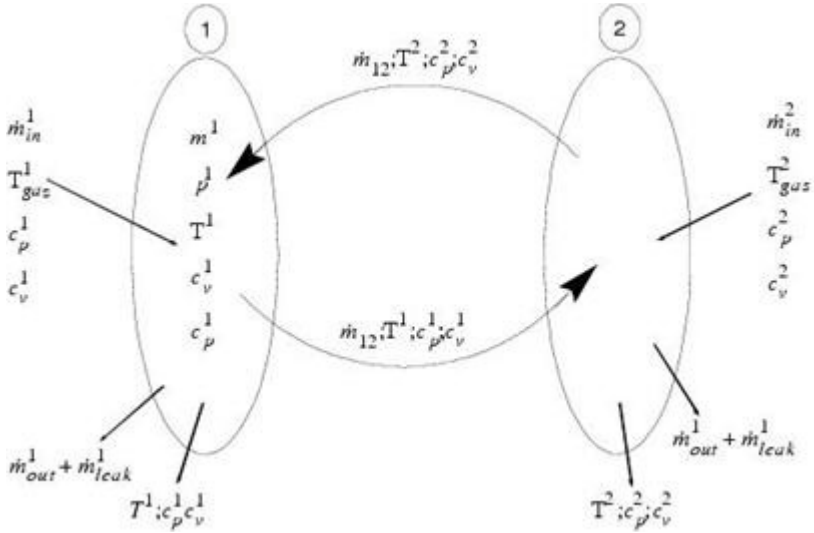
Field	Contents		Type	Default
ID	Number of the GBAGC entry		I > 0	Required
GID1	Number of a GBAG entry		I > 0	Required
GID2	Number of a GBAG entry, different from GID1		I > 0	Required
TRIGGER	The time-dependent parameters are offset in time.		C	TIME
	TIME	The offset is defined at TRIGGERV.		
TRIGGERV	The value of the offset in time		R	Required
PRESTOL	If the pressure difference between the two gas bags is less than this value, no mass flow occurs. The value is specified as a percentage.		R ≥ 0.0	0.0
CD12	The variation of the discharge coefficient for the opening allowing flow from gas bag 1 into gas bag 2.		C	CONSTANT
	CONSTANT	The discharge coefficient is constant and is specified in CD12V.		
	TABLE	The discharge coefficient varies with pressure. CD12V is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with the pressure (x-value).		
	TIME	The area varies with time. CD12V is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		

Field	Contents		Type	Default
CD12V	The discharge coefficient or the number of a TABLED1 or TABLEEX entry depending on the value of CD12. Discharge coefficients must be between zero and one.		R or I > 0	1.0
A12	The variation of the total area of the opening that allows flow from gas bag 1 into gas bag 2.		C	CONSTANT
	CONSTANT	The area is constant and specified in A12V.		
	TABLE	The area varies with pressure. A12V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value).		
	TIME	The area varies with time. A12V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
A12V	The total area of the opening or the number of a TABLED1 or TABLEEX entry, depending on the value of A12.		R or I > 0	0.0
CD21	The variation of the discharge coefficient for the opening that allows flow from gas bag 2 into gas bag 1.		C	CONSTANT
	CONSTANT	The discharge coefficient is constant and is specified in CD21V.		
	TABLE	The discharge coefficient varies with pressure. CD21V is the number of a TABLED1 or TABLEEX entry giving the variation of discharge coefficient (y-value) with the pressure (x-value). The discharge coefficient must be between zero and one.		
	TIME	The discharge coefficient varies with time. CD21V is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
CD21V	The discharge coefficient or the number of a TABLED1 or TABLEEX entry, depending on the value of CD21.		R or I > 0	1.0
A21	The variation of the total area of the opening that allows flow from gas bag 2 into gas bag 1.		C	CONSTANT
	CONSTANT	The area is constant and specified in A21V.		

Field	Contents		Type	Default
	TABLE	The area varies with pressure. A21V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value).		
	TIME	The area varies with time. A21V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
A21V	The total area of the opening or the number of a TABLED1 or TABLEEX entry, depending on the value of A21.		R or I > 0	0.0

Remarks

1. Both gbags are triggered before flow between the two gas bags begins.
2. The energy balance and mass flow is as shown in the following graphic:



3. One GBAG can be referenced in multiple GBAGC entries.
4. For compartmented air bags, you can model each compartment as a separate gas bag and connect the gas bags using GBAGC entries.
5. The GBAGC entry is obsolete. It is preferable to model connections between GBAG entries using the GBAG → GBAGPOR → PORFGBG logic. See *Dytran User's Guide*, Chapter 6: Air Bags and Occupant Safety, Porosity in Air Bags for more details.

GBAGCOU General Coupling to Gas Bag Switch

Defines a switch from full gas dynamics to uniform pressure formulation.

Format and Example

1	2	3	4	5	6	7	8	9	10
GBAGCOU	ID	CID	GID	TSTART	PERCENT				
GBAGCOU	1	100	101	0.0	5				

Field	Contents	Type	Default
ID	Unique number of a GBAGCOU entry	I > 0	Required
CID	Number of a COUPLE entry	I > 0	Required
GID	Number of a GBAG entry	I > 0	Required
TSTART	Time after which the coupling algorithm checks if a switch to the uniform pressure method is valid. It is valid when the following is true:	R > 0.0.	0.0
	$Max\left[\frac{(P_{max}-P_{average})}{P_{average}}, \frac{(P_{average}-P_{min})}{P_{average}}\right] < \frac{PERCENT}{100}$		
	where		
	P_{max} = maximum Eulerian pressure exerted on the SURFAE		
	P_{min} = minimum Eulerian pressure exerted on the SURFACE		
	$P_{average}$ =average Eulerian pressure exerted on the SURFAE		
PERCENT	Value used in validity check as defined above.	R > 0.0	5%

Remarks

1. The SURFACE SID referenced by the COUPLE entry CID and by the GBAG entry GID must be equal.
2. All Eulerian and general coupling calculations are deactivated after transition from gas dynamics to uniform pressure.

GBAGHTR Heat Transfer Model to be used With GBAG Entry

Defines the heat transfer model to be used with GBAG.

Format and Example

1	2	3	4	5	6	7	8	9	10
GBAGHTR	CID	HTRID	SUBID	HTRTYPE	HTRTYPID	COEFF	COEFFV		
GBAGHTR	101	83		HTRCONV	2	TABLE	14		

Field	Contents		Type	Default
CID	Unique number of a GBAGHTR entry		I > 0	Required
HTRID	Number of a set of GBAGHTR entries HTRID must be referenced from a GBAG entry		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the GBAG entry.	I ≥ 0	0
	= 0	GBAGHTR definitions are used for the entire SURFACE referred to from the GBAG entry.		
HTRTYPE	Defines the type of heat transfer.		C	
	HTRCONV	The HTRCONV logic is used to model heat transfer through convection in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is transported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 exposes the complete subsurface area, while a value of COEFFV = 0.0 results in no heat transfer through the subsurface.		
	HTRRAD	The HTRRAD logic is used to model heat transfer through radiation in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is transported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 exposes the complete subsurface area, while a value of COEFFV = 0.0 results in no heat transfer through the subsurface.		
COEFF	Method of defining the area coefficient.		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified on COEFFV.		

Field	Contents		Type	Default
	TABLE	The area coefficient varies with time. COEFFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0 < R < 1 or I > 0	1.0

Remarks

1. A combination of multiple GBAGHTRs with different HTRTYPEs is allowed.
2. All options of HTRTYPE can also be referenced by a COUHTR. It allows for setting up the exact same model for either a uniform pressure model (GBAG to GBAGHTR) or an Eulerian model (COUPLE to COUHTR). It is then possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).
3. For the same SUBSURF multiple, different types of heat transfer may be defined.
4. A more detailed description can be found in *Dytran User's Guide*, Chapter 6: Air Bags and Occupant Safety, Porosity in Air Bags for more details.

GBAGINFL

Inflator Model to be used with GBAG Entry

Bulk Data Entry Descriptions defines an inflator model suited for air bag analyses using the uniform pressure approach (GBAG). The inflator model is defined as part of the GBAG surface.

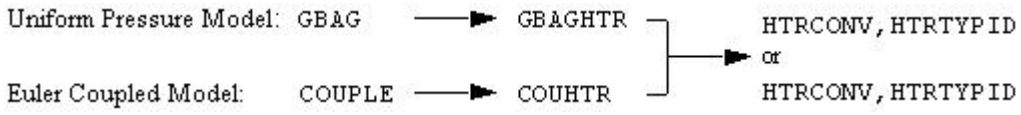
Format and Example

1	2	3	4	5	6	7	8	9	10
GBAGINFL	CID	INFID	SUBID	INFTYPE	INFTYPID	COEFF	COEFFV		
GBAGINFL	201	1	120	INFLHYB	11		0.012		

Field	Contents		Type	Default
CID	Unique number of a GBAGINFL entry		I > 0	Required
INFID	Number of a set of GBAGINFL entries NFID must be referenced from a GBAG entry.		I > 0	Required
SUBID	Number of a SUBSURF, which must be a part of the SURFACE referred to from the GBAG entry.		I > 0	Required
INFTYPE	Defines the type of inflator.		C	Required
	INFLATR	The INFLATR logic is used to model inflators in an air bag.		
	INFLATR1	The INFLATR1 logic is used to model inflators in an air bag.		
	INFLHYB	The INFLHYB logic is used to model hybrid inflators in an air bag.		
	INFLHYB1	The INFLHYB1 logic is used to model hybrid inflators in an air bag.		
	INFLCG	The INFLCG models a cold gas inflator.		
INFTYPID	Number of the entry selected under INFTYPE, for example, INFLATR, INFTYPID.		I > 0	Required
COEFF	Method of defining the area coefficient.		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified on COEFFV.		
	TABLE	The area coefficient varies with time. COEFFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0. < R < 1. or I > 0	1.0

Remarks

1. The INFLATR, INFLATR1, INFLHYB, or INFLHYB1 inflator geometry and location is defined by a subsurface (SUBID). The area of the hole through which the gas enters is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 opens up the complete subsurface area, while a value of COEFFV = 0.0 results in a closed inflator area with no inflow.
2. The same INFTYPE entry referenced from this GBAGINFL entry can be referenced by a COUINFL entry. This allows for setting up the exact same model for either a uniform pressure model or an Euler Coupled model:



   This makes it possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).
3. One GBAG entry can reference more than one COUINFL entry. This allows for modeling of multiple inflators in an air bag module.

GBAGPOR

Gas Bag Porosity

Defines the porosity model to be used with GBAG.

Format and Example

1	2	3	4	5	6	7	8	9	10
GBAGPOR	CID	PORID	SUBID	PORTYPE	PORTYPID	COEFF	COEFFV		
GBAGPOR	7	100	365	PERMEAB	63		0.99		

Field	Contents		Type	Default
CID	Unique number of a GBAGPOR entry		I > 0	Required
PORID	Number of a set of GBAGPOR entries		I > 0	Required
	PORID must be referenced from a GBAG entry.			
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the GBAG entry.	I ≥ 0	0
	= 0	GBAGPOR definitions are used for the entire SURFACE referred to from the GBAG entry.		
PORTYPE	Defines the type of porosity			
	PORHOLE	The PORHOLE logic is used to model holes in an air bag. The hole is defined by a subsurface (SUBID). The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 opens up the complete hole area, while a value of COEFFV = 0.0 results in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORHOLE entry, with ID as defined on the PORTYPID.		

Field	Contents		Type	Default
	PERMEAB	The PERMEAB logic is used to model permeable air-bag material. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference between the gas inside the air bag and the environmental pressure. The function is specified on a PERMEAB entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		
	PORFGBG	The PORFGBG logic is used to model gas flow through a hole in the coupling surface connected to a GBAG. The hole is defined by a subsurface (SUBID). The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 opens up the complete hole area, while a value of COEFFV = 0.0 results in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORFGBG entry, with ID as defined on the PORTYPID.		
	PERMGBG	The PERMGBG logic is used to model gas flow through a permeable area in the coupling surface connected to a GBAG. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference. This function is specified on a PERMGBG entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		
COEFF	Method of defining the porosity coefficient		C	CONSTANT
	CONSTANT	The porosity coefficient is constant and specified on COEFFV.		

Field	Contents		Type	Default
	TABLE	The porosity coefficient varies with time. COEFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The porosity coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0.0 < R < 1.0 or I > 0	

Remarks

1. The combination of multiple GBAGPORs with different PORTYPEs is allowed.
2. All options of PORTYPE can also be referenced by a COUPOR. It allows for setting up the exact same model for either a uniform pressure model (GBAG to GBAGPOR) or an Eulerian model (COUPLE to COUPOR). It is then possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).
3. The options PORFGBG and PERMGBG can be used to model air bags with different compartments.

GRAV

Gravity Field

Defines a gravity acceleration field.

Format and Example

1	2	3	4	5	6	7	8	9	10
GRAV	LID		SCALE	NX	NY	NZ			
GRAV	4		-2.0	0.	1.				

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
SCALE	Gravity scale factor	R	1.0
NX, NY, NZ	Components of gravity vector. At least one component must be nonzero.	R	0.0

Remarks

1. The gravity acceleration $g(t)$ is defined as

   $$\bar{g}(t) = SCALE \cdot \bar{N} \cdot T(t)$$

   where $SCALE$ is the gravity scale factor; $\bar{N}$ is the vector defined by NX, NY, and NZ; and $\bar{T}(t)$ is the value interpolated at time t from the table referenced by the TLOADn entry.
2. LID must be referenced by a TLOADn entry.
3. The type field on the TLOADn entry must be set to zero.
4. One gravitational field can be defined per problem.
5. The gravitational accelerations are applied to all masses in the problem.

GRDSET

Grid Point Default

Defines default options for the GRID entries.

Format and Example

1	2	3	4	5	6	7	8	9	10
GRDSET		CP					PS		
GRDSET							3456		

Field	Contents	Type	Default
CP	Number of a coordinate system in which the location of the grid point is defined.	I ≥ 0	0
PS	Single-point constraints associated with the grid point. This should be an integer of any of the digits 1 through 6.	I > 0	0

Remarks

1. Any GRID entry with a blank value of PS is set to the value given on this entry. Note that the constraints on the GRID and GRDSET entries are not cumulative; i.e., if there is a GRDSET entry with constraint code 34 and a GRID entry with constraint code 2, the grid point is constrained only in direction 2.
2. There can only be one GRDSET entry in the input data.

GRID

Grid Point

Defines the location of a geometric grid point in the model and its constraints.

Format and Example

1	2	3	4	5	6	7	8	9	10
GRID	ID	CP	X	Y	Z		PS		
GRID	2		1.0	-2.0	3.0		316		

Field	Contents	Type	Default
ID	Unique grid-point number	I > 0	Required
CP	Number of a coordinate system in which the location of the grid point is defined.	I ≥ 0	See Remark 2.
X, Y, Z	Location of the grid point	R	0.0
PS	Permanent single-point constraints associated with the grid point. This must be an integer made up of the digits 1 through 6 with no embedded blanks.	I > 0	See Remark 2.

Remarks

1. All grid-point numbers must be unique.
2. If CP or PS is blank or zero, the value given on the GRDSET entry is used.
3. Grid points can also be constrained using the SPC and SPC1 entries.
4. The values of X, Y and Z depend on the type of the coordinate system CP. Their meaning in each type of coordinate system is listed below.

Type	X	Y	Z
Rectangular	X	Y	Z
Cylindrical	R	θ	Z
Spherical	R	θ	ϕ

θ and ϕ are measured in degrees.

GROFFS

Grid Point Offset

Defines a grid-point offset in the global coordinate system.

Format and Example

1	2	3	4	5	6	7	8	9	10
GROFFS	ID	SID	XOFF	YOFF	ZOFF				
GROFFS	32	2	8.E-4	0.75	0.0				

Field	Contents	Type	Default
ID	Unique grid-point offset number.	I > 0	Required
SID	Number of a SET1 entry containing a list of grid points.	I > 0	Required
XOFF, YOFF, ZOFF	Components of a vector defining the grid-point offset. The offset is in the global coordinate system regardless of the CP defined in the GRIDoption.	R	0.0

HGSUPPR

Hourglass Suppression Method

Defines the hourglass suppression method and the corresponding hourglass damping coefficients.

Format and Example

1	2	3	4	5	6	7	8	9	10
HGSUPPR	HID	PROP	PID	HGTYPE	HGCMEM	HGCWRP	HGCTWS	HGCSOL	+
HGSUPPR	1	SHELL	100	FBV	0.1	0.1	0.1		+
+C	RBRCOR	VALUE							
+	YES	10000							

Field	Contents		Type	Default
HID	Hourglass suppression definition number		I > 0	Required
PROP	Property type		C	See Remark 1.
PID	Property number		I > 0	See Remark1.
HGTYPE	Hourglass suppression method:		C	See Remark 2.
	FBV	for shells only (default for shells)		
	FBS	for solids only (default for solids)		
	Dyna	for shells only		
HGCMEM	Membrane damping coefficient		0.0 ≤ R ≤ 0.15	0.1
HGCWRP	Warping damping coefficient		0.0 ≤ R ≤ 0.15	0.1
HGCTWS	Twisting damping coefficient		0.0 ≤ R ≤ 0.15	0.1
HGCSOL	Solid damping coefficient		0.0 ≤ R ≤ 0.15	0.1
RBRCOR	Rigid body rotation correction:		C	NO
	NO	No rigid-body rotation correction is applied to hourglass resisting forces.		
	YES	Rigid-body rotation correction is applied to hourglass resisting forces.		
	See Remark 3.			
VALUE	Number of steps		R > 0.0	See Remark 3.

Remarks

1. The property type definition and the property number are required. Since property numbers are unique within a certain class of element types, the property type and the property number uniquely define to what elements the hourglass suppression method and coefficients apply. The following property types are valid entries:

BAR	For bar elements
BEAM	For beam elements
BELT	For belt elements
COMP	For composite shell elements
DAMP	For damper elements
ELAS	For spring elements
EULER	For Eulerian elements
ROD	For rod elements
SHELL	For shell elements
SOLID	For solid Lagrangian elements

   It must be noted however, that only shell CQUAD4 and Lagrangian CHEXA and CPENTA elements can suffer from undesired hourglass modes. All HGSUPPR entries referring to other types of elements are ignored.

2. There are three types of hourglass suppression methods available in Dytran. These are standard DYNA viscous (DYNA) hourglass damping, the Flanagan-Belytschko Stiffness (FBS) hourglass damping for solids, and the Flanagan-Belytschko Stiffness (FBV) hourglass damping for shells.

   Lagrangian solid elements can address FBS suppression; shell elements can address DYNA and FBV suppression. The default for the Lagrangian solid elements is FBS. The default for the shell elements is FBV.

   The default hourglass suppression method can be globally changed by the PARAM, HGTYPE.

3. The rigid-body rotation correction on the hourglass forces is only necessary in cases where shell elements undergo a large rigid-body rotation. If the RBRCOR field is set to YES, and the VALUE field is left blank, the correction is applied during each time step. If the VALUE field is set to a number, the rotation correction is applied only when the rigid-body rotation would result in a rotation of the element over 90° in less than VALUE time steps. Usually, if the rigid-body rotation correction is necessary; 10000 is a good value. This option saves some CPU time.

   The RBRCOR option applies to the Key-Hoff shell formulation only; for all other element types and formulations, the option is ignored.

4. The membrane, warping and twisting coefficients apply to shell elements only; for all other element types, the data is ignored. The solid damping coefficient applies to solid Lagrangian elements only; for all other element types, the data is ignored.

   The default value of the damping coefficients can be globally changed by PARAM , HGCOEFF.

5. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression.

HTRCONV

Air Bag Convection

Defines the heat transfer through convection for a COUPLE and/or GBAG (sub)surface.

Convection is heat transfer from the air bag to the environment through the air bag surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
HTRCONV	HTRID	HTRCF-C	HTRCF-T	TENV					
HTRCONV	8		14	293.0					

Field	Contents	Type	Default
HTRID	Unique number of a HTRCONV entry	I > 0	Required
HTRCF-C	Constant heat transfer convection coefficient	R > 0.0	See Remark 2.
HTRCF-T	The heat transfer convection coefficient is a tabular function of time. The number given here is the number of a TABLED1 or TABLEEX entry.	I > 0	See Remark 3.
TENV	Environmental temperature	R > 0.0	Required

Remarks

1. The HTRCONV entry can be referenced from a COUHTR and/or GBAGHTR entry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state. Two of the four gas constants (γ, R, c_v and/or c_p) have to be defined on the EOSGAM entry.
3. Either HTRCF-C or HTRCF-T must be specified.
4. Energy only transfers out of the air bag if the temperature inside the air bag is higher than the environmental temperature.
5. A more detailed description of heat transfer by convection can be found in *Dytran User's Guide*, Chapter 6: Air Bags and Occupant Safety, Heat Transfer in Air Bags.

HTRRAD

Air Bag Radiation

Defines the heat transfer through radiation for a COUPLE and/or GBAG (sub)surface.

Radiation is heat transfer from the air bag to the environment through the air bag surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
HTRDAD	HTRID	GASEMI-C	GASEMI-T	TENV	SBOLTZ				
HTRRAD	2	0.15		293.0	5.676E-8				

Field	Contents	Type	Default
HTRID	Unique number of a HTRRAD entry	I > 0	Required
GASEMI-C	Constant gas emissivity	R > 0.0	See Remark 3.
GASEMI-T	The gas emissivity is a tabular function of time. The number given here is the number of a TABLED1 or TABLEEX entry.	I > 0	See Remark 3.
TENV	Environmental temperature	R > 0.0	Required
SBOLTZ	Stephen-Boltzmann constant	R > 0.0	Required

Remarks

1. The HTRRAD entry can be referenced from a COUHTR and/or GBAGHTR entry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state. Two of the four gas constants (γ, R, c_v and/or c_p) have to be defined on the EOSGAM entry.
3. Either GASEMI-C or GASEMI-T must be specified.
4. Energy only transfers out of the air bag if the temperature inside the air bag is higher than the environmental temperature.
5. A more detailed description of heat radiation can be found in *Dytran User's Guide*, Chapter 6: Air Bags and Occupant Safety, Heat Transfer in Air Bags.

HYDSTAT

Hydrostatic Preset of Density in Euler Elements

Initializes the Euler element densities in accordance to a hydrostatic pressure profile.

Format and Example

1	2	3	4	5	6	7	8	9	10
HYDSTAT	HID	MID	GID	CID	XCG	YCG	ZCG	PATM	
HYDSTAT	101	4			0	0	0	100000	

Field	Contents	Type	Default
HID	Identification number of the HYDSTAT entry	I > 0	Required
MID	Material to which the hydrodynamic pressure profile will be applied	I > 0	Required
GID	Number of a grid point at the free surface	I > 0	See Remark 4.
CID	Local coordinate system	I > 0	See Remark 4.
CXG, YCG, ZCG	Coordinates of a point at the free surface	R	See Remark 4.
PATM	Pressure at free surface	R	Required

Remarks

1. It is assumed that each Euler domain contains at most two Eulerian materials and includes the GRAV option. One material has to be a fluid using EOSPOL; the other a gas or void. This EOSPOL material is given by the MID entry. The interface between gas and fluid is the free surface and is assumed to be normal to the gravity vector as specified on the GRAV entry. For example if the gravity vector points in the z-direction then the interface between the gas and the fluid has to be horizontal.
2. The hydrostatic preset changes the density of the fluid like material in order to conform to the hydrostatic preset. It overrules the material densities as specified on the TICEL and TICVAL options. Densities of the gas like material are not changed.
3. The free surface has to match with material fractions as defined in the initialization of Euler elements by the TICEL and TICEUL options. The hydrostatic preset only changes densities; it does not change material fractions.
4. There are two options to enter the location of the free surface. The first option is to enter a grid point number. In that case, the CID and XCG-ZCG fields must be left blank. The gridpoint entry already has the option of using a local coordinate system. When coordinates are used, the GID field has to be left blank.
5. If there is no structural grid point indicating the free surface, a new grid point can be defined that is only used for determining the free surface level. If the gravity vector points in the z-direction, only the z-coordinate of the grid point is used. The x and y coordinate can be chosen arbitrarily. Similar remarks hold when the gravity vector is in one of the other coordinate directions. The same holds when using coordinates instead of a grid point.

6. PATM should be equal to the pressure in the air.
7. When coupling surfaces are present, the HYDSTAT ID needs to be referenced by at least one coupling surface. For each coupling surface, a different HYDSTAT entry can be defined. Several COUPLE options can refer to the same HYDSTAT ID. If no HYDSTAT ID is specified on a COUPLE entry, the Euler elements associated to this coupling surface are not initialized with a hydrostatic preset.

IGNORE

Ignore a Set of Euler Elements

Defines an interaction between a coupling surface and a set of Euler elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
IGNORE	IID	CSID	SETID						
IGNORE	1	1	1						

Field	Contents	Type	Default
IID	Unique ignore number	I > 0	Required
CSID	Coupling surface ID	I > 0	Required
SETID	Set1 ID	I > 0	Required

Remarks

1. The coupling surface will interact with all Euler elements except the ones defined by the SET1 referring to the SETID.
2. This option can only be used in combination with PARAM, FASTCOUP.

INCLUDE

Starts Reading a New Input File

Switches reading of the input data to another file. Once that file is read, processing reverts back to the original file immediately after the INCLUDE statement.

Format and Example

1	2	3	4	5	6	7	8	9	10
`INCLUDE file name`									
`INCLUDE BULK.DAT`									

Field	Contents	Type	Default
`filename`	Name of the new input file name to be used. The name must be appropriate for the machine that is executing Dytran.	C	No new file

Remarks

1. The file must be present in the area where Dytran is executing.
2. A comma cannot be used to separate the fields.
3. BEGIN BULK and ENDDATA can be present in the `INCLUDE` file.

INFLATR

Air Bag Inflator Model

Defines the inflator characteristics of a couple and/or GBAG subsurface.

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLATR	INFLID	MASFLR-T	TEMP-T	TEMP-C	γ,GASNAM	c_v	R	c_p	
INFLATR	5	100		907.0		283.0			
+	MID								
+	1								

Field	Contents	Type	Default
INFLID	Unique number of an INFLATR entry	I > 0	Required
MASFLR-T	Table number of a TABLED1 or TABLEEX entry specifying the mass flow rate as a function of time.	I > 0	Required
TEMP-T	Table number of a TABLED1 or TABLEEX entry specifying the dynamic temperature of the inflowing gas as a function of time.	I > 0	See Remark 3.
TEMP-C	Constant value of the dynamic temperature of the inflowing gas constant.	R > 0.0	See Remark 3.
γ, GASNAM	Ratio of specific heat constants if real. Name of an INFLGAS entry if character.	R > 0.0 or C	See Remark 4.
c_v	Specific heat at constant volume	R > 0.0	See Remark 5.
R	Gas constant	R > 0.0	See Remark 5.
c_p	Specific heat at constant pressure	R > 0.0	See Remark 5.
MID	Material number of the inflator material	1 > 0	See Remark 2. Used only for MMHYDRO solver.

Remarks

1. The INFLATR entry can be referenced from a COUINFL and/or GBAGINFL entry.
2. When used in combination with the single material hydrodynamic Euler solver, an EOSGAM (ideal gas) equation of state is required. In that case, the material number, MID, can be left blank. When using the Multi-material solver, the material number, MID, has to point to one of the Eulerian materials and the equation of state of that material has to be of type EOSGAM.

3. Either TEMP-C or TEMP-T must be specified. The INFLATR entry uses the specified temperature as the dynamic temperature of the inflowing gas. The dynamic temperature is the temperature of the moving gas, as opposed to the static temperature which is also known in literature as total-, rest-, or stagnation temperature and refers to the temperature of the gas when brought to rest from its moving condition. The INFLATR1 entry uses the static temperature of the inflowing gas.
4. If the γ, GASNAM field contains a real entry real or is left blank, the inflator gas constants are given on the INFLATR entry itself, see Remark 5. Otherwise, the entry is read as the name of an INFLGAS entry. In this case, the remaining entries must be left blank.
5. Specify only two of the four gas constants. They are related as:

$$\gamma = \frac{c_p}{c_v} \text{ , } R = c_p - c_v$$

INFLATR1

Air Bag Inflator Model

Defines the inflator characteristics of a couple and/or GBAG subsurface.

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLATR1	INFLID	MASFLR-T	TEMP-T	TEMP-C	γ, GASNAM	c_v	R	c_p	
INFLATR1	5	100		907.0		283.0			
+	MID								
+	1								

Field	Contents	Type	Default
INFLID	Unique number of an INFLATR1 entry.	I > 0	Required
MASFLR-T	Table number of a TABLED1 or TABLEEX entry specifying the massflow-rate as a function of time.	I > 0	Required
TEMP-T	Table number of a TABLED1 or TABLEEX entry specifying the static temperature of the inflowing gas as a function of time.	I > 0	See Remark 3.
TEMP-C	Constant value of the static temperature of the inflowing gas.	R > 0.0	See Remark 3.
γ, GASNAM	Ratio of specific heat constants if real. Name of an INFLGAS entry if character.	R > 0.0 or C	See Remark 4.
c_v	Specific heat at constant volume.	R > 0.0	See Remark 5.
R	Gas constant.	R > 0.0	See Remark 5.
c_p	Specific heat at constant pressure	R > 0.0	See Remark 5.
MID	Material number of the inflator material	1 > 0	See Remark 2. Used only for MMHYDRO solver.

Remarks:

1. The INFLATR1 entry can be referenced from a COUINFL and/or GBAGINFL entry.
2. When used in combination with the single material hydrodynamic Euler solver, an EOSGAM (ideal gas) equation of state is required. In that case, the material number, MID, can be left blank. When using the Multi-material solver, the material number, MID, has to point to one of the Eulerian materials and the equation of state of that material has to be of type EOSGAM. The Multi-material solver does not allow the use of gas fractions.

3. Either TEMP-C or TEMP-T must be specified. The INFLATR1 entry uses the specified temperature as the static temperature of the inflowing gas. In literature the static temperature is also known as total-, rest-, or stagnation temperature and refers to the temperature of the gas when brought to rest from its moving condition, as opposed to the dynamic temperature which refers to the temperature of the moving gas. The INFLATOR entry uses the dynamic temperature of the inflowing gas.
4. If the γ, GASNAM field contains a real entry real or is left blank, the inflator gas constants are given on the INFLATR1 entry itself (see Remark 5.). Otherwise, the entry is read as the name of an INFLGAS entry. In this case, the remaining entries must be left blank.
5. Specify only two of the four gas constants. They are related as:

$$\gamma = \frac{c_p}{c_v} \quad R = c_p - c_v$$

INFLCG

Airbag Cold Gas Inflator Model

Defines the cold gas-inflator characteristics of a couple and/or GBAG subsurface.

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLCG	INFLID	TANKVOL	INITPRES	INITTEMP	INITMAS	γ, GASNAM	c_v	R	+INFLC1
INFLCG	111	0.857	131325	293	1.37	1.14		286	+INFLC1
+INLC1	c_p								+
+INLC1									+

Field	Contents	Type	Default
INFLID	Unique number of an INGLCG entry	I > 0	Required
TANKVOL	Tank volume	R > 0.0	Required
INITPRES	Initial tank pressure	R > 0.0	See Remark 3.
INITTEMP	Initial tank temperature	R > 0.0	Required
INITMAS	Initial gas mass of inflator	R > 0.0	See Remark 3.
γ, GASNAM	Ratio of specific heat constants if real. Name of an INFLGAS entry if character. See Remark 4.	R > 0.0 or C	See Remark 5.
c_v	Specific heat at constant volume	R > 0.0	See Remark 6.
R	Gas constant	R > 0.0	See Remark 6.
c_p	Specific heat at constant pressure.	R > 0.0	See Remark 6.

Remarks

1. The INFLTANK entry can be referenced from a COUNIFL and/or GBAGINFL entry.
2. When used in an Euler coupled analysis, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state.
3. Either INITPRES or INITMAS has to be specified, but not both. The relation between INITMAS and INITPRES is given by:

$$INITPRES = R\frac{IINITMAS}{TANKVOL}INITTEMP$$

4. The cold gas inflator is a reservoir filled with high pressure gas. It is assumed that the volume stays constant at TANKVOL. The mass inside the inflator will steadily decrease due to flow into the Euler domain or to a GBAG. Due to inertia, it can happen that the pressure of the inflator becomes less than the outside pressure. In that case, some inflow into the inflator occurs. Transport between inflator and the Euler domain or GBAG is based on the constancy of total temperature. This is equivalent to the pressure method.
5. If this field contains a real entry real or is left blank, the inflator gas constants are given on the INFLATR1 entry itself (see Remark 5.). Otherwise, the entry will be read as the name of an INFLGAS entry. In this case, the remaining entries must be left blank.
6. Specify only two of the four gas constants. They are related as:

$$\gamma = \frac{c_p}{c_v} \qquad R = c_p - c$$

INFLFRAC — Hybrid Inflator Gas Fraction Definition

Specifies mass fractions or molar fractions as a function of time for a hybrid inflator definition.

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLFRAC	FRACID	TYPE	TIMEID	TIME	FRAC1	FRAC2	FRAC3	-etc.-	
INFLFRAC	99	MASS	TIME	0.0	0.15	0.0	0.55		

Field	Contents		Type	Default
FRACID	Unique number of an INFLFRAC entry		I > 0	Required
TYPE	Specifies whether mass fractions or molar fractions is given:		C	MASS
	MASS	The fractions on INFLFRAC are mass fractions.		
	MOLAR	The fractions on INFLFRAC are molar fractions. See Remark 6.		
TIMEID	Defines a new line of data		C	Required
	TIME	Specifies that data for a new time increment is given. See Remark 7.		
TIME	Time for which the gas fractions are given		R ≥ 0.0	Required
FRACi	Fraction of gas i at the specified time		R ≥ 0.0	See Remark 8.

Remarks

1. The INFLFRAC entry must be referenced from an INFLHYB or INFLHYB1 entry.
2. Fraction values of the inflowing gas is linearly interpolated between the specified time increments.
3. Use as many continuation lines as necessary to completely define the gas fractions. The data for a time step are preceded by a TIME keyword. Missing entries are set to 0.
4. The order of the gases for which the fractions are specified is identical to the order in which the gases are specified on the INFLHYB or INFLHYB1 entry.
5. At least one line of gas fractions must be given.
6. If molar fractions are to be used, the universal gas constant must be specified through PARAM,UGASC.
7. At least one of the fractions for each time step must have a value greater than 0.0.
8. Fractions for each time step should add up to 1.0. If this is not the case, they are scaled so that they do.

INFLGAS

Inflator Gas Definition

Defines a thermatically ideal gas to be used with a standard or hybrid inflator.

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLGAS	GASNAM	TYPE	VALUE	CPGAS	V1	V2	V3	V4	
INFLGAS	CO2	RSPEC	189.	CONSTANT	846				

Field	Contents		Type	Default
GASNAM	Unique name of an INFLGAS entry		C	Required
TYPE	Specific gas constant or molar weight specified		C	RSPEC
	RSPEC	Specific gas constant		
	MOLWT	Molar weight; see Remark 2.		
VALUE	Value of the variable TYPE		R > 0.0	Required
CPGAS	The variation of the specific heat constant at constant pressure.		C	CONSTANT
	CONSTANT	The specific heat is constant and specified in V1.		
	TABLE	The specific heat constant is temperature-dependent. V1 is the number of a TABLED1 entry giving the variation of the specific heat with the temperature.		
	POLY	The specific heat constant is temperature-dependent. V1 through V4 are the coefficients of a polynomial expression; see Remark 3.		
V1	The specific heat constant, the number of aTABLED1 entry or the first polynomial coefficient, depending on the value of CPGAS.		R or I > 0	Required
V2,V3,V4	Coefficients of polynomial expression when CPGAS equals POLY.		R	0.0

Remarks

1. INFLGAS can be referenced by an INFLATR,INFLATR1, INFLHYB, INFLHYB1, or INITGAS entry.
2. When the molar weight is given, the universal gas constant R_{uni} must be specified using PARAM, UGASC, so that:

   $$R_{spec} = R_{uni}/MOLWT$$

3. A polynomial expression for cp is given by:

$$\Gamma) = V1 + V2 \cdot T + V3 \cdot T^2 +$$

4. The specific heat constant at constant volume cv is calculated from the specific heat constant at constant pressure cp, the universal gas constant and the molecular weight according to:

$$c_v = c_p(T) - R_{spec}$$

5. The ratio of specific heats is given as:

$$\gamma = c_p / c_v$$

INFLHYB

Hybrid Inflator Model

Defines the hybrid-inflator characteristics of a couple and/or GBAG subsurface.

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLHYB	INFLHID	MASFLR-T	TEMP-T	TEMP-C	FRAC				+
INFLHYB	9	15		650.	12				+
+	GASNAM1	GASNAM2	GASNAM3	-etc.-					
+	22	25	3						

Field	Contents	Type	Default
INFLID	Unique number of an INFLHYB entry	I > 0	Required
MASFLR-T	Table number of a TABLED1 or TABLEEX entry specifying the mass flow rate as a function of time.	I > 0	Required
TEMP-T	Table number of a TABLED1 or TABLEEX entry specifying the dynamic temperature of the inflowing gas as a function of time.	I > 0	See Remark 3.
TEMP-C	Constant value of the dynamic temperature of the inflowing gas constant.	R > 0.0	See Remark 3.
FRAC	Number of an INFLFRAC entry specifying the fractions of the inflowing gas as a function of time.	I > 0	Required
GASNAMi	Name of an INFLGAS entry	C	See Remark 4.

Remarks

1. The INFLHYB entry can be referenced from a COUINFL and/or GBAGINFL entry.
2. When used in an Euler coupled analysis, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state.
3. Either TEMP-C or TEMP-T must be specified. The INFLHYB entry uses the specified temperature as the dynamic temperature of the inflowing gas. The dynamic temperature is the temperature of the moving gas, as opposed to the static temperature which is also known in literature as total-, rest-, or stagnation temperature and refers to the temperature of the gas when brought to rest from its moving condition. The INFLHYB1 entry uses the static temperature of the inflowing gas.
4. At least one inflator gas must be specified using an INFLGAS entry. There is no limit to the number of inflator gases per INFLHYB.

INFLHYB1

Hybrid Inflator Model

Defines the hybrid-inflator characteristics of a couple and/or GBAG subsurface.

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLHYB1	INFLHID	MASFLR-T	TEMP-T	TEMP-C	FRAC				+
INFLHYB1	9	15		650.	12				+
+	GASNAM1	GASNAM2	GASNAM3	-etc.-					
+	22	25	3						

Field	Contents	Type	Default
INFLID	Unique number of an INFLHYB1 entry	I > 0	Required
MASFLR-T	Table number of a TABLED1 or TABLEEXentry specifying the massflow-rate as a function of time.	I > 0	Required
TEMP-T	Table number of a TABLED1 or TABLEEXentry specifying the static temperature of the inflowing gas as a function of time.	I > 0	See Remark 3.
TEMP-C	Constant value of the temperature of the inflowing gas	R > 0.0	See Remark 3.
FRAC	Number of an INFLFRAC entry specifying the fractions of the inflowing gas as a function of time.	I > 0	Required
GASNAMi	Name of an INFLGASentry	C	See Remark4.

Remarks

1. The INFLHYB1 entry can be referenced from a COUINFL and/or GBAGINFL entry.
2. When used in an Euler coupled analysis, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state.
3. Either TEMP-C or TEMP-T must be specified. The INFLHYB1 entry uses the specified temperature as the static temperature of the inflowing gas. In literature the static temperature is also known as total-, rest-, or stagnation temperature and refers to the temperature of the gas when brought to rest from its moving condition as opposed to the dynamic temperature that refers to the temperature of the moving gas. The INFLHYB entry uses the dynamic temperature of the inflowing gas.
4. At least one inflator gas must be specified using an INFLGAS entry. There is no limit to the number of inflator gases per INFLHYB1.

INFLTANK

Air Bag Tanktest Inflator Model

Defines the Tanktest-inflator characteristics of a couple and/or GBAG subsurface

Format and Example

1	2	3	4	5	6	7	8	9	10
INFLTANK	INFLID	METH-TANK	TPTABLE	TANKVOL	INFLMAS	INITPRES	ENDTPRES	INITTEMP	+
INFLTANK	111	AVTEMP	10	0.12	0.01	0.0			+
+	ENDTEMP	γ	c_v	R	c_p	IPTABLE	INFLPRES	INFLTEMP	+
+		1.4		286.					+
+	INFLAREA	SFTP	SFIP						
+									

Field	Contents		Type	Default
INFLID	Unique number of an INFLATANK entry		I > 0	Required
METH-TANK	Method of calculating the mass-flowrate:		C	Required
	AVTEMP	Average Temperature Method		
	INFPRES	Inflator Pressure Method		
TPTABLE	Table number of a TABLED1 or TABLEXentry specifying the tank pressure as a function of time.		I > 0	Required
TANKVOL	Tank volume		R > 0.0	Required
INFLMAS	Total gas mass generated by inflator		R > 0.0	Required
INITPRES	Initial tank pressure		R > 0.0	Required. See Remark 3.
ENDPRES	End tank pressure		R > 0.0	Required. See Remarks 4.and 5.
INITTEMP	Initial tank temperature		R > 0.0	Required. See Remark 5.
ENDTEMP	End tank temperature		R > 0.0	Required.See Remark 5.
γ	Ratio of specific heat constants		R > 0.0	See Remark 7.
c_v	Specific heat at constant volume		R > 0.0	See Remark 7.
R	Gas constant		R > 0.0	See Remark 7.
c_p	Specific heat at constant pressure		R > 0.0	See Remark 7.

Field	Contents		Type	Default
IPTABLE	Table number of a TABLED1 or TABLEEX entry specifying the inflator pressure as a function of time.		I > 0	Required. See Remark 5.
INFLPRES	Initial inflator pressure		R > 0.0	Required. See Remarks 5. and 6.
INFLTEMP	Temperature of inflowing gas:		R > 0.0 or C	ATM. See Remark 5.
	ATM	Use average temperature of AVTEMP method		
	Real value	User specified temperature		
INFLAREA	Total area of inflator holes		R > 0.0	Required. See Remark 5.
SFTP	Scale factor for tank pressure		R > 0.0	1.0. See Remark 5.
SFIP	Scale factor for inflator pressure		R > 0.0	1.0. See Remark 5.

Remarks

1. The INFLTANK entry can be referenced from a COUINFL and/or GBAGINFL entry.
2. When used in an Euler coupled analysis, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state.
3. The initial tank pressure entry (INITPRES) is interpreted as an absolute pressure and used to define reference pressure at t=0 in the tank. The different between INITPRES and the pressure value at t=0 from the table will be added to the entire pressure curve of TPTABLE.
4. The end tank pressure entry (ENDPRES) is interpreted as an absolute pressure at t=tend of tank pressure table (TPTABLE). This value is used for calculation of total generated mass in the tank.
5. This field must be specified only when Inflator Pressure Method (INFPRES) is defined in the METHTANK field.
6. The initial inflator pressure entry (INFLPRES) is interpreted as an absolute pressure and used to define reference pressure at $t = 0$ in the inflator. The different between INFLPRES and the pressure value at $t = 0$ from the table is added to the entire pressure curve of IPTABLE.
7. Specify only two of the four gas constants. They are related as:

$$\gamma = \frac{c_p}{c_v} \quad R = c_p - c_v$$

INITGAS Gas Bag or Coupling Surface Initial Gas Fraction Definition

Specifies the initial gas composition inside a gas bag or Euler coupling surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
INITGAS	INTID		GASNAM1	FRAC1	GASNAM2	FRAC2	-etc.-		
INITGAS	4		CO2	0.4	O2	0.11			

Field	Contents	Type	Default
INTID	Unique number of an INITGAS entry	I > 0	Required
GASNAMi	Name of an INFLGAS entry	C	See Remark 3.
FRACi	Mass fraction of gas i.	R ≥ 0.0	See Remark 4.

Remarks

1. The INITGAS entry can be used to specify the initial gas composition for a gas bag or for an Eulerian coupling surface. The INTID must be referenced either from a GBAG cad or a COUPLE entry.
2. Use as many continuation lines as necessary to completely define the gas fractions.
3. At least one INFLGAS reference must be given.
4. Fractions should add up to 1.0. If this is not the case, they will be scaled so that they do.

JOIN Hinge-type Join of Six DOF Grid Points with Three DOF Grid Points

Defines a hinge-type join of Lagrangian elements with six degrees of freedom grid points (for example CHEXA CQUAD4, CBAR etc.) to Lagrangian elements with three degrees of freedom grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
JOIN	ID	SID	TOL						
JOIN	1	100	1.E-6						

Field	Contents	Type	Default
ID	Unique JOIN number	I > 0	Required
SID	Number of a SET1 entry containing the list of grid points to be joined	I > 0	Required
TOL	Tolerance for joining the grid points. Grid points that have mutual distance within this tolerance are joined.	R ≥ 0	5.10-4

Remarks

1. Grid points with the same number of degrees of freedom (DOF) can be equivalenced in the preprocessing phase.
2. The JOIN gives rise to a hinge connection. A stiff connection can be achieved by using KJOIN.

KJOIN Kinematic Join of Six DOF Grid Points with Three DOF Grid Points

Defines the joining of Lagrangian elements with six degrees of freedom grid points (for example, CHEXA CQUAD4, CBAR, etc.) to Lagrangian elements with three degrees of freedom grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
KJOIN	ID	SID	TOL	INTERFERE	STIFFNESS				
KJOIN	ID	150	1.E-5	STRONG	0.5				

Field	Contents	Type	Default
ID	Unique KJOIN number	I > 0	Required
SID	Number of a SET1 entry containing the list of grid points to be joined	I > 0	Required
TOL	Tolerance for joining the grid points. Grid points with mutual distance that is within the tolerance are joined.	R > 0.0	5.E–4
INTERFERE	Defines whether the rotation present at a six DOF grid point interferes with the rotation from the kinematic constraint (STRONG or NONE).	C	STRONG
STIFFNESS	Defines the relative stiffness of the kinematic join.	R	0.0

Remarks

1. To change the stiffness of the join, the STIFFNESS field can be defined.
2. Stiffness is increased by setting INTERFERE to none.
3. The kinematic join acts as a locally inserted stiff element.
4. The STIFFNESS field defines a relative stiffness where the value should be in the interval (-1/2, 1/2). Values less than zero reduce the stiffness, and values greater than zero increase the stiffness.
5. Geometric aspects are automatically taken into account.
6. In cases where the set of grid points for the KJOIN is too large to fit in one SET1 entry, you can define multiple SET1 entries with the same set number. The SET1 entries that have the same set number are automatically merged into one set.
7. You can define a hinge connection by using the JOIN entry.

MAT1

Material Property Definition, Form 1

Defines the material properties for linear, isotropic materials.

Format and Example

1	2	3	4	5	6	7	8	9	10
MAT1	MID	E	G	NU	RHO				
MAT1	17	3.+7		0.33	4.28				

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
E	Young's modulus E	R ≥ 0.	See Remark 2.
G	Shear modulus G	R ≥ 0	See Remark 2.
NU	Poisson's ratio ν	0. < R ≤ 0.5	See Remark 4.
RHO	Mass density ρ	R > 0.0	Required

Remarks

1. The material number must be unique for all MAT1 and MAT8 entries.
2. The following rules apply when E, G, or ν are blank:
   a. E and G cannot both be blank.
   b. If ν and E or ν and G are both blank, then both are set to 0.0.
   c. If only one of E, G, or ν is blank, it is computed from the equation:
   $$E = 2(1+\nu)G$$
3. Implausible data on one or more MAT1 entries results in a User Warning Message. Implausible data is defined as any of the following:
   $$E < 0.0 \text{ or } G < 0.0 \text{ and } \nu > 0.5 \text{ or } \nu < 0.0.$$
4. It is strongly recommended that only two of the values be specified on the MAT1 entry.

MAT2 Anisotropic Material for Shells

Defines anisotropic material for shells. The shell cross-section properties are constant throughout the analysis.

Format and Example

1	2	3	4	5	6	7	8	9	10
MAT2	MID	G11	G12	G13	G22	G23	G33	RHO	
MAT2	7	7.088e6	3.435e6	3.229e6	4.218e6	3.229e6	3.556e6	1.31e-4	

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
GIJ	The material property matrix	R	Required. See Remark 2.
RHO	Mass density	R	See Remark 3.

Remarks

1. This material can **only** be used in combination with a PSHELL entry to activate the classical lamination theory for shell-structure analysis. If this material is used as transverse shear, then only G11, G12, and G22 are read. Other material data is ignored. This option is only available for C0-TRIA, BLT and KEYHOFF shell formulation.

   The convention for the GIJ in fields 3 through 8 are represented by the matrix relationship as follows:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{12} & G_{22} & G_{23} \\ G_{13} & G_{23} & G_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix}$$

2. The mass density RHO is required in case of a membrane material definition.
3. No sublayer variables are available these elements. The element outputs are the stress resultants (NXX, NYY, NXY, MXX, MYY, MXY, QYZ, and QZX).

MAT8 Orthotropic Elastic Material Properties

Defines the properties for an orthotropic material for shell elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
MAT8	MID	E1	E2	N12	G12	G1,Z	G2,Z	RHO	
MAT8	171	30.+6	1.+6	0.3	2.+6	3.+6	1.5+6	0.056	

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
E1	Modulus of elasticity in longitudinal direction (also defined as fiber direction or one-direction).	R > 0.0	Required
E2	Modulus of elasticity in lateral direction (also defined as matrix direction or two-direction).	R > 0.0	Required
N12	Poisson's ratio (ε2/ε1 for uniaxial loading in one-direction). Note that ν21 = ε1/ε2 for uniaxial loading in two-direction is related to ν12, E1, E2 by the relation ν12 E2 = ν21 E1.	R > 0.0	Required
G12	In-plane shear modulus	R > 0.0	Required
G1,Z	Transverse shear modulus for shear in 1-Z plane (default implies G1,Z = G12).	R > 0.0	Blank
G2,Z	Transverse shear modulus for shear in 2-Z plane (default implies G2,Z = G12).	R > 0.0	Blank
RHO	Mass density	R > 0.0	Required

Remarks

1. An approximate value for G1,Z and G2,Z is the in-plane shear modulus G12. If test data is not available to accurately determine G1,Z and G2,Z if the material and transverse shear calculations are deemed essential, the value of G12 may be supplied for G1,Z and G2,Z. The MSC Nastran defaults for G1,Z and G2,Z are infinite if left blank. Dytran assumes the transverse shear moduli to be equal to G12.
2. Excess data as defined in the MSC Nastran MAT8 continuation lines is ignored. Equivalent entries can be defined in the MAT8A Bulk Data entry.
3. This material model can only be referenced from a PCOMP entry.

MAT8A Orthotropic Failure Material Properties

Defines the failure properties for an orthotropic material for shell elements.

Example in FMS Section of the Dytran input stream for FT=USER1:

```
CONNECT SERVICE mymat SCA.MDSolver.Obj.Uds.Dytran.Materials
```

Format and Example

1	2	3	4	5	6	7	8	9	10
MAT8A	MID	FT	NV	S	ALPHA	TRSFAIL	F12	GROUP	+
MAT8A	7	COMBINAT		100.					+
+	XT	XC	YT	YC	PFD	VALUE	PFDST		+
+	200	150	100	110.	STEPS	200			+
+	FBTEN	FBCOM	MXTEN	MXCOM	MXSHR				+
+	CHANG	STRSS	MODTSAI	MODTSAI	STRSS				+
+									+
+									+
+	PRDFT	PRDFC	PRDMT	PRDMC	PRDSH				
+					0011				

Field	Contents		Type	Default
MID	Unique material number		I	See Remark 1.
FT	Failure theory to be used to test whether the element layer fails. See Remark 2.		C	Blank
	Blank	No failure		
	HILL	Tsai-Hill theory		
	TSAI	Tsai-Wu theory		
	MODTSAI	Modified Tsai-Wu theory		
	STRSS	Maximum stress		
	CHANG	Chang-Chang theory		
	USER1	User-defined model that allows for property degradation	C	See Remark 8.
	COMBINAT	Combination	C	See Remark 6.

Field	Contents		Type	Default
	HASHIN	Hashin theory	C	
NV	Number of additional history variables for a user model. See Remark 8.		0 < I < 1000	0
S	Failure stress for in-plane shear		R > 0.0	See Remark 4.
ALPHA	Nonlinear shear coefficient. See Remark 5.		R ≥ 0.0	0.
TRSFAIL	Transverse shear failure.		C	SUBL
	ELEM	Failure if element fails		
	SUBL	Failure if sublayer fails		
F12	Interaction term in Tsai-Wu theory		R	0.
GROUP	The group name used for the FMS section CONNECT SERVICE statement. It is used only when FT is set to USER1. See Remark 8.		C	Blank
XT, XC	Tensile compressive failure stress in the large structural direction		R > 0.0	See Remark 4.
YT, YC	Tensile compressive failure stress in the lateral direction		R > 0.0	See Remark 4.
PFD	Post-failure degradation model. See Remark 9.		C	STEPS
	STEPS	Degrade stresses by time steps		
	TIME	Degrade stresses by time		
	VELOC	Degrade stresses by velocity		
VALUE	Depending on PFD, VALUE gives the number of time steps, time interval, or propagation velocity.		I or R	100
PFDST	Post-failure degradation start.			
	INDV	Stresses are degraded per distinct failure mode.	C	INDV
	ALL	Stresses are degraded if all elastic constants are zero.		
	See Remark 9..			
FBTEN, FBCON, MXTEN, MXCOM, MXSHR	Failure modes in fiber, matrix direction, and theory failure. Enter values if FT = COMBINAT		C	See Remark 6.
PRDFT	Property degradation due to fiber-tension failure		I	1111
PRDFC	Property degradation due to fiber-compression failure		I	1010
PRDMT	Property degradation due to matrix-tension failure		I	0110

Field	Contents	Type	Default
PRDMC	Property degradation due to matrix-compression failure	I	0110
PRDSH	Property degradation due to in-plane shear failure	I	0001

Remarks

1. The material number must refer to a MAT8 material definition.
2. If a failure theory is selected other than USER1 or COMBINAT, the theory defines the following failure modes:

CHANG	Fiber tension, matrix tension/compression
HILL	All modes
TSAI	All modes
MODTSAI	Matrix tension/compression
STRSS	All modes
HASHIN	Fiber tension/compression
	Matrix tension/compression

   For an element to fail completely, both fiber and matrix in all sublayers must fail.
3. This material model can only be referenced from a PCOMP entry.
4. Failure stresses are required if a failure theory is selected.
5. ALPHA is used for all failure theories to define a nonlinear stress-strain relation.
6. The individual failure modes are defined according to the corresponding mode in the theory as listed under FT. To be relevant, the theory must define the failure mode (see Remark 2.). You must enter data if FT is set to COMBINAT.
7. The property degradation rules due to the various failure modes are listed below:

Material Constant	Failure Mode				
	Fiber Tens	Fiber Comp	Matrix Tens	Matrix Comp	Shear
E1	X	X			
E2	X		X	X	
V12	X	X	X	X	
G12	X				X

The Poisson's ratio Nu21 is treated the same as Nu12.

To override the default model, an integer value is defined as a packed word in the following order:

$(E_1)\ (E_2)\ (Nu12)\ (G12)$

1 denotes property degradation.

0 denotes no degradation.

The last five fields of the MAT8A Bulk Data entry are input for the user to specify the degradation behavior for each mode of failure.

8. NV is required input and NV new user variables are automatically created. User variables for sublayers are used on restart and archive output. Refer to them as USRnLxx where n is the user ID and xx is the sublayer number (see Dytran User's Guide, Chapter 9: Running the Analysis, Running Dytran Using the dytran Command). The values S, XT, XC, YT, and YC are also required input when FT is set to USER1. For the model USER1, usrExcomp (cpp) and ext_comp.F (Fortran) user routines have to be used.
9. The PFD entry indicates how the stresses are degraded to zero. The PFDST indicates when the stresses start to degrade.

   Using ALL means that degradation starts when all material constants (E_1, E_2, $Nu12$, G_{12}) are degraded to zero as specified by the FT entry and the property degradation rules. Note that property degradation means that the stress increments are zero but that the stresses degrade according to PFD.

   INDV means that stress degradation starts for the fiber stress if $E_1 = 0$, for matrix stress if $E_2 = 0$, and for shear stress if $G_{12} = 0$.
10. Any failure theory introduces five additional sublayer variables. The PFDST entry introduces three additional variables. The number of user variables is defined by NV.

MATINI

Eulerian Initialization Surface

Defines a surface that is used for initialization of regions inside an Eulerian mesh with user-defined initial conditions.

Format and Example

1	2	3	4	5	6	7	8	9	10
MATINI	CID	SID	COVER	REVERSE	CHECK				
MATINI	100	37							

Field	Contents		Type	Default
CID	Unique number of an MATINI entry		I > 0	Required
SID	Number of a SURFACE entry defining the initialization surface		I>0	Required
COVER	The processing strategy for Eulerian elements inside and outside of the initialization surface.		C	INSIDE
	INSIDE	The part of the Eulerian elements that lie inside the closed volume of the initialization surface will obtain the initial conditions belonging to that surface.		
	OUTSIDE	The part of the Eulerian elements that lie outside the closed volume of the initialization surface obtains the initial conditions that belong to that surface		
REVERSE	Auto reverse switch for MATINI surface segments:		C	ON
	ON	If necessary, the normals of the MATINI surface segments are automatically reversed so that they all point in the same general direction and give a positive closed volume.		
	OFF	The segments normals are not automatically reversed.		
CHECK	Checking switch for MATINI surface segments:		C	ON
	ON	The normals of the segments are checked to see whether they all point in the same general direction and give a positive closed volume.		
	OFF	The segment normals are not checked.		
	When REVERSE is set to ON, CHECK is automatically set to ON.			

Remarks

1. All initialization surfaces must form a multifaceted closed volume.
2. In case the surface is defined as a set of segments attached to shell elements you must define the elements as dummy elements by choosing the DUMMY option on the FORM entry of the PSHELL1 entry.
3. An initialization surface can only be used to initialize regions in an Eulerian mesh with appropriate initial conditions. An initialization surface can not be used as a coupling surface, contact surface or as a rigid surface.
4. The normals of all the segments that form the initialization surface must point in the same general direction and results in a positive closed volume. Setting the REVERSE option to ON ensures that this condition is satisfied, regardless of how the segments are defined initially.
5. The COVER option determines how Eulerian elements that are (partially) inside or outside of the initialization surface are processed.

MATRIG

Rigid-Body Properties

Defines the properties of a rigid body.

Format and Example

1	2	3	4	5	6	7	8	9	10
MATRIG	MID	RHO	E	NU	MASS	XC	YC	ZC	+
MATRIG	7	7850.	210.E9	0.3	750	0.0	7.0	-3.0	+
+	IXX	IXY	IXZ	IYY	IYZ	IZZ	CID	COG-FL	+
+	17.0	13.2	14.3	20.9	15.7	10.0	12		+
+	VX	VY	VZ	WX	WY	WZ			+
+			13.3						+
+	XC-LOCAL	YC-LOCAL	ZC-LOCAL						+
+									+
+	VELMAX	DSPMAXN	DSPMAXP	XDIR	YDIR	ZDIR			
+			10.0	0.0	0.0	1.0			

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
RHO	Density	R > 0.0	1.0
E	Young's modulus	R > 0.0	1.0
NU	Poisson's ratio	0.0 ≤ R < 0.5	0.2
MASS	Mass of the rigid body	R > 0.0	See Remark 2.
XC,YC,ZC	x, y, and z coordinates of the center of gravity	R	See Remark 7.
IXX, IXY, IXZ, IYY, IYZ, IZZ	Inertia tensor of the rigid body about the center of gravity	R	See Remark 7.
CID	Number of a coordinate system in which the inertia tensor and the center of gravity are defined.	I > 0	See Remarks 8. and 11.

Field	Contents	Type	Default
COG-FL	Flag to indicate recalculation of Center of Gravity of a rigid body. For symmetry models, recalculation of the Center of Gravity is not desired. YES Recalculate the Center of Gravity of the rigid body. NO Do not recalculate the Center of Gravity of the rigid body when XC, YC, or ZC are not blank.	C	YES
VX, VY, VZ	Initial translational velocity of the center of gravity in the basic coordinate system.	R	0.0
WX, WY, WZ	Initial rotational velocities of the rigid body about the center of gravity in the basic coordinate system.	R	0.0
XC-LOCAL YC-LOCAL, ZC-LOCAL	x, y, and z local coordinates of the center of gravity	R	See Remark 11.
VELMAX	Limiting the maximum magnitude of the translational velocity.	R≥0.0	See Remark 12.
DSPMAXN	Limiting the maximum of the translational displacement in the negative direction of the rigid body control.	R≥0.0	See Remark 12.
DSPMAXP	Limiting the maximum of the translational displacement in the positive direction of the rigid body control.	R≥0.0	See Remark 12.
XDIR, YDIR, ZDIR	Vector indicating the positive direction of the translational rigid body control.	R	See Remark 12.

Remarks

1. All coordinates are defined in the basic coordinate system.
2. If MASS is blank or zero, the mass is calculated from the density and the geometry of the mesh defining the rigid body.
3. The continuation lines are not required.
4. The MATRIG definition is used instead of a DYMATn definition and is referenced by properties PSOLID, PSHELL, PBAR, and PBEAMn. Different properties can refer to the same MATRIG entry forming one rigid body. The matrmerg or matrmrg1 option (see PARAM,MATRM(E)RG(1)) can be used for merging different MATRIG and RBE2-FULLRIG definitions into one single rigid body.
5. By using PARAM,RBE2INFO,GRIDON, the grid points of the MATRIG will be listed in the output file.
6. If the fields VX, VY, VZ, WX, WY, and WZ are blank, the initial conditions of the rigid body are calculated from the initial velocities on the TIC and TIC1 entries referring to grid points attached to the rigid body. The net initial conditions are the average of those for all the grid points attached to the rigid body.

If the initial conditions are set using the VX, VY, VZ, WX, WY, and WZ fields, the TIC and TIC1 entries referring to grid points attached to the rigid body are ignored.

7. If the inertia tensor or the coordinates of the center of gravity are undefined, then they are computed from the density or mass and the geometry of the mesh defining the rigid body.
8. The inertia tensor can only be defined in a local rectangular coordinate system. If the entry for a local coordinate system is left blank, then the inertia tensor is defined in the global coordinate system.
9. The behavior of rigid bodies in discussed in Lagrangian Elements.
10. The mass and the location of the Center of Gravity for a rigid body can change from the input cards defined by the user when the rigid body is attached to elements with elastic-plastic materials. The node that is shared will get a mass from both the rigid body and from the mass of the elastic-plastic element. For correct rigid body mechanical, the added mass of the node from the elastic body needs to be added to the total mass of the rigid body. To this end, also the location of the center of gravity might change location. However, in cases the rigid model was modeled with symmetry planes, then it is not desired to recalculate the Center of Gravity. The extra mass of the elastic-plastic element is always added to the mass of the rigid body.
11. The center of gravity can be defined in a local rectangular coordinate system (CID). However, XC, YC, and ZC (x, y, and z coordinates of the center of gravity in the basic coordinate system) should be left blank when XC-LOCAL, YC-LOCAL, ZC-LOCAL (x, y, and x coordinates of the center of gravity in a local coordinate system) are defined.
12. The maximum translational motion of a rigid body can be restricted by the optional VELMAX, DSPMAXN, and DSPMAXP entries. Blank entries correspond to no motion control, while 0.0 entries set the limiting displacement and velocity values to 0.0. DSPMAXN and DSPMAXP limit the maximum displacement of a rigid body in positive and negative directions of a local axis defined by (XDIR, YDIR, ZDIR). If either DSPMAXN or DSPMAXP is specified, then the (XDIR, YDIR, ZDIR) must be defined. This feature is designed primarily to control the motion of rigid tooling in metal forming applications, so it must be used with caution in other scenarios. For constraining rigid body motion in general applications, the users are encouraged to use RBC3 or TLOAD1 (TYPE=12) instead.

MCOHE Interface Cohesive Zone Modeling Element Material Properties

Specifies material cohesive properties for a fully nonlinear element used to simulate the onset and progress of delamination based on Turon at el (2006).

Format and Example

1	2	3	4	5	6	7	8	9	10
MCOHE	MID	MODEL							
MCOHE	3	1							
+	COHE	CRTOD	MAXOD	SNSR				SFC	
+	2.8E-1	4.52E-4	2.154E-2	3.0				2.0	
+	SNER	ETA	RHO						
+	2.821	1.45	0.1						

Field	Contents	Type	Default
MID	Identification number of a MCOHE1 entry.	I > 0	Required.
MODEL	Cohesive material type 1 Bilinear model	I > 0	1
COHE	Cohesive energy, facture toughness in Mode I.	R > 0.0	Required.
CRTOD	Critical opening distance at the start of damage in Mode I.	R > 0.0	Required.
MAXOD	Maximum opening distance at failure in Mode I.	R > 0.0	Required.
SNSR	Shear to normal stress ratio (Mode II vs Mode I).	R > 0.0	1.0
SFC	Stiffness factor in compression in Mode I.	R > 0.0	1.0
SNER	Shear to normal energy ratio (Mode II vs Mode I).	R > 0.0	1.0
ETA	Coupling coefficient for mixed-modes of Mode I and Mode II	R	1.0
RHO	Density.	R > 0.0	Required.

Remarks

1. Maximum normal stress can be calculated by critical energy in Mode I and failure normal displacement, δ_N^f,

$$\tau_{Nmax} = \frac{2GIc}{\delta_N^f}$$

where,

τ_{Nmax} : Maximum normal stress

GIc : Cohesive energy, facture toughness in Mode I,

δ_N^f : MAXOD = Maximum opening distance at failure in Mode I.

Maximum tangential stress, τ_{Tmax} and Cohesive energy, facture toughness in Mode II, $GIIc$ are

$$\tau_{Tmax} = SNSR \cdot \tau_{Nmax}$$

$$GIIc = SNER \cdot GIc$$

where,

SNSR: Shear to normal stress ratio,

SNER: Shera to normal energy ratio.

Initial normal stiffness, K_N^0 can be calculated from

$$K_N^0 = \frac{\tau_{Nmax}}{\delta_N^0}$$

where,

δ_N^0 : CRTOD = Critical opening distance at the start of damage in Mode I

Initial tangential stiffness, K_T^0 is calculated by

$$K_T^0 = K_N^0 \frac{GIc}{GIIc}\left(\frac{\tau_{Tmax}}{\tau_{Nmax}}\right)^2$$

Initial damage displacements in Mode II, δ_T^0 is

$$\delta_T^0 = \frac{\tau_{Tmax}}{K_T^0}$$

Failure damage displacements in Mode II, δ_T^f is

$$\delta_T^f = \frac{GIIc}{\tau_{Tmax}}$$

Mixed mode displacement of Mode I and Mode II, δ_B is

$$\delta_B = \frac{K_N^0 \delta_N^2 + K_T^0 \delta_T^2}{\sqrt{(K_N^0 \delta_N)^2 + (K_T^0 \delta_T)^2}}$$

The ratio of energy release rate in tangential to total energy release rate, *B* is

$$B = \frac{K_T^0 \delta_T^2}{K_N^0 \delta_N^2 + K_T^0 \delta_T^2}$$

Mixed mode stiffness, K_B is

$$K_B = K_N^0 (1 - B) + K_T^0 B$$

Based on equations above, mixed mode initial damage displacement, δ_B^0 and δ_B^f are calculated by

$$\delta_B^0 = \sqrt{\frac{K_N^0 \delta_N^{0\,2} + (K_T^0 \delta_T^{0\,2} - K_N^0 \delta_N^{0\,2}) B^\eta}{K_B}}$$

$$\delta_B^f = \frac{K_N^0 \delta_N^0 \delta_N^f + (K_T^0 \delta_T^0 \delta_T^f - K_N^0 \delta_N^0 \delta_N^f) B^\eta}{K_B \delta_B^0}$$

where,

η : Coupling coefficient for mixed-modes of Mode I and Mode II.

The damage variable, *d* is

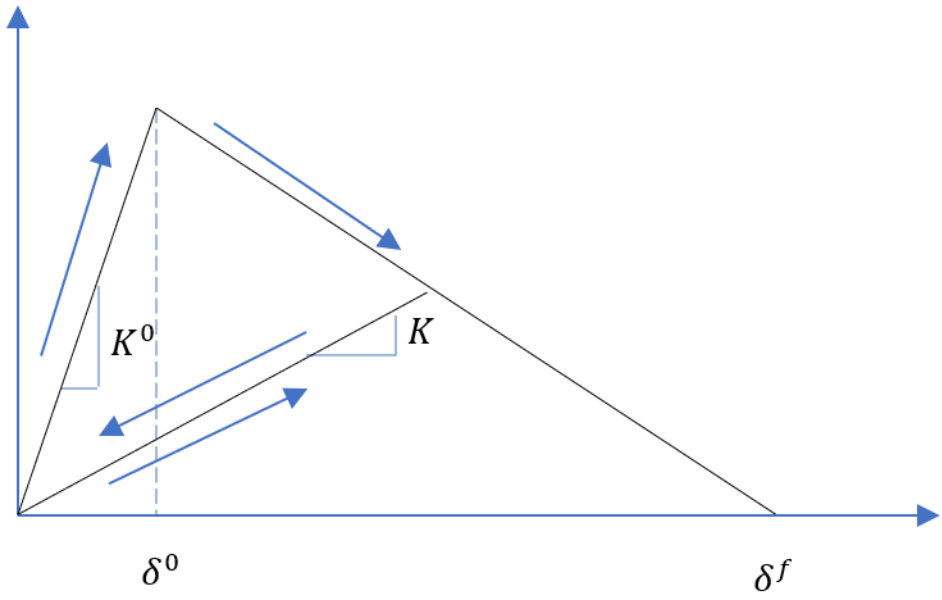
$$0.0 \le d = \frac{\delta_B^f (\delta_B - \delta_B^0)}{\delta_B (\delta_B^f - \delta_B^0)} \le 1.0$$

The stiffnesses in normal and tangential direction, K_N and K_T are set to

$$K_N = (1-d)K_N^0$$

$$K_T = (1-d)K_T^0$$

The element will be deleted when d = 1.0.



Loading, unloading and reloading route under damage

In compression, to avoid penetration, the stiffness can be increased by SFC in normal direction.

$$K_{comp} = SFC \cdot K_N^0$$

Reference

- Turon, Albert et al. "A damage model for the simulation of delamination in advanced composites under variable-mode loading", *Mechanics of Materials,* 38(11), pp.1072-1089 (2006).
- Benzeggagh, M.L. and Kenane, M., "Measurement of mixed-mode delamination fracture toughness of unidirectional glass/epoxy composites with mixed-mode bending apparatus", Composites Science and Technology, 56, 439-449 (1996).

MESH

Mesh Generator

Defines a mesh.

Format and Example

1	2	3	4	5	6	7	8	9	10
MESH	MID	TYPE	DXEL	DYEL	DZEL	XREF	YREF	ZREF	+
MESH	1	ADAPT	0.1	0.2	0.3				+
+	X0	Y0	Z0	DX	DY	DZ			+
+									+
+	NX	NY	NZ	SUBMESH	NSTGP	NSTEL	PROP	PID	+
+							EULER	101	+
+	RESIZE	TID-X	TID-Y	TID-Z	METHOD				
+	SCALE	101			ALL				
+	BIAS	GROWFX	GROWFY	GROWFZ	IBIDX	IBIDY	IBIDZ		
+	CENTER	1.2	1.2	1.2					
	XOBX	YOBX	ZOBX	DXBX	DYBX	DZBX			
	0	1	0	10	50	100			
+	CID								
+	102								
+	NELCUBE	NBX	NBY	NBZ	PROCDIR	NPX	NPY	NPZ	
+									

Field	Contents		Type	Default
MID	Unique MESH number		I > 0	Required. See Remark 1.
TYPE	Type of mesh generation.		C	Required. See Remark 1.
	ADAPT	An Euler mesh will be created around a coupling surface. This option is only valid for PROP=EULER and requires that the MID of the MESH is referenced from the MESHID of the COUPLE entry. During the simulation, when the coupling surface moves or deforms, the Euler mesh adapts itself by adding and removing elements. The adapt algorithm ensures that the coupling surface is contained inside the Euler mesh at all times with the minimum amount of elements. The Euler elements are aligned with the basic coordinate.		
	BOX	Rectangular mesh aligned with the basic coordinate system is created, filled with HEXA elements. This option can be used for PROP=EULER and PROP=SOLID.		
DEXEL, DYEL, DZEL	Euler element sizes		R	See Remark 1.
XREF, YREF, ZREF	Coordinates of reference point:		R	-1e-6
	For TYPE=ADAPT, these coordinates provide control over the location of the Euler mesh, to avoid that the faces of the Euler mesh are initially at the same location as faces of the coupling surface.			
	For TYPE=BOX, these coordinates will be used as the original of the mesh. They are the default setting for X0, Y0, and Z0.			

Field	Contents		Type	Default
X0,Y0,Z0	Coordinates of point of origin.		R	XREF, YREF, ZREF
	Not used for TYPE=ADAPT.			
DX,DY,DZ	Width of mesh in different directions.		R	See Remark 1.
	For TYPE=ADAPT, these values will only be used if DXEL, DYEL, and DZEL are left blank.			
NX,NY,NZ	Number of elements in the different directions		I > 0	See Remark 1.
	For TYPE=ADAPT, these values will only be used if DXEL, DYEL, and DZEL are left blank.			
SUBMESH	Allows using smaller mesh sizes for a part of the mesh. SUBMESH is the MESH ID of a finer mesh that is to replace part of the mesh.			See Remark 10.
NSTGP	Starting grid-point number		I > 0	See Remark 2.
	Not used for TYPE=ADAPT.			
	If there are multiple couple surfaces then the starting grid-point number can only be specified if PARAM, FLOW-METHOD, FACET has been activated.			
NSTEL	Starting element number		I > 0	See Remark 2.
	Not used for TYPE=ADAPT			
	If there are multiple couple surfaces then the starting element number can only be specified if PARAM, FLOW-METHOD, FACET has been activated.			
PROP	Property type:		C	Required
	EULER	An Eulerian mesh is created.		
	SOLID	A Lagrangian mesh is created.		
PID	Property number.		I > 0	Required
	For PROP=EULER, this number references a PEULER or PEULER1.			
	For PROP=SOLID, this number references a PSOLID.			
RESIZE	Only valid for TYPE=ADAPT		C	None
	Option to change the element size during the simulation:			
	NONE	No resizing of DX, DY, DZ during simulation.		

Field	Contents		Type	Default
	SCALE	The elements are resized by a scale-factor as a function of time.		
	LENGTH	The elements are resized by specifying the length as a function of time.		
TID-X	ID of a TABLED1 See RESIZE for contents of table.		1 > 0	Blank
	It must define a step function. See Remarks 4.and 5.			
TID-Y	ID of a TABLED1 See RESIZE for contents of table.		1 > 0	TID-X
	It must define a step function. See Remarks 4. and 5.			
TID-Z	ID of a TABLED1 See RESIZE for contents of table.		1 > 0	TID-X
	It must define a step function. See Remarks 4.and 5.			
METHOD	Method for determining when to create Euler elements:		C	ALL; used only for resizing
	ALL	Always remesh an existing Euler element. Maintains existing void regions.		
	MATERIAL	Only remesh those Euler elements that contain material. Removes void regions.		
	See Remark 7.			
BIAS	Adds bias to the mesh			
	CENTER	Starting at the center of the BOX, the mesh size gradually changes such that the mesh size at the boundaries of the BOX is GROWX times the mesh size at the center.		

Field	Contents		Type	Default
	REF	Starting at (X0, Y0, Z0), the mesh size gradually changes such that the mesh size at the boundaries of the BOX is GROWX times the mesh size at the center.		
GROWX, GROWY, GROWZ	Total grow factor; the ratio between the finest and the coarsest element size.		R > 0.0	Required only if BIAS is not blank.
IBIDX, IBIDY, IBIDZ	ID-numbers of bias entries		I > 0	See Remarks 9 and 10.
X0BX, Y0BX, Z0BX, DXBX, DYBX, DZBX	For adaptive Euler meshes, there are two methods to generate Euler archives.		R	See Remarks 11 and 12.
	■ There is only output for currently existing elements. Consequently the geometry changes and for each cycle a new Euler archive is written.			
	■ By defining an auxiliary box. All adaptive elements that are within the box for one of the cycles requested are stored in the archive. This allows multiple cycles in one Euler archive. This box should be sufficiently large such that it contains all elements. The fields X0BX, Y0BX, and Z0BX specify the start point and DXBX, DYBX, and DZBX specify the with of box and. If the there are adaptive elements outside, the box a warning is given.			
CID	ID - number of a local coordinate system		I ≥ 0	0
NELCUBE	The number of elements per cube. This number is used as a guideline. The actual number sued per cube can differ and can be found in the out file.			See Remark 12.
NBX	Overrules NELCUBE. The number of cubes in the x-direction.			
NBY	The number of cubes in the y-direction.			
NBZ	The number of cubes in the z-direction.			

Field	Contents		Type	Default
PROCDIR	This directive controls the way cubes are distributed across processors. The effect can be checked by checking the Eulerian output variable PARTITION.		C	X
	X	Partition in global x-direction first.		
	Y	Partition in global y-direction first.		
	Z	Partition in global z-direction first.		
NPX	The number of partitions in the x-direction. Required for PROCDIR=USER.		I	1
NPY	The number of partitions in the y-direction.		I	NPX
NPZ	The number of partitions in the z-direction.		I	NPX

Remarks

1. The grid points of the mesh are generated at the following locations:

   Type=ADAPT:

   $$(x, y, z) = (XREF + i * DXEL, YREF + j * DYEL, ZREF + k * DZEL)$$

   Grid points and elements located a certain distance outside the coupling surface are not created. This saves memory and CPU time.

   When (XREF, YREF, ZREF) are outside the coupling surface, no actual grid point is created at this location, but the mesh is shifted appropriately.

   Type=BOX:

   $$(x, y, z) = (X0 + i * DXEL, Y0 + j * DYEL, Z0 + k * DZEL)$$

   Nodes and elements are always created, even if the MESH is referenced from the MESHID of a COUPLE entry.

   One of the following input combinations is required:

   TYPE=ADAPT

   a. DXEL, DYEL, DZEL or

   b. DX, DY, DZ and NX, NY, NZ

      →DXEL = DX/NX; DYEL = DY/NY; DZEL = DZ/NZ

   TYPE=BOX

   a. DXEL, DYEL, DZEL and NX, NY, NZ or

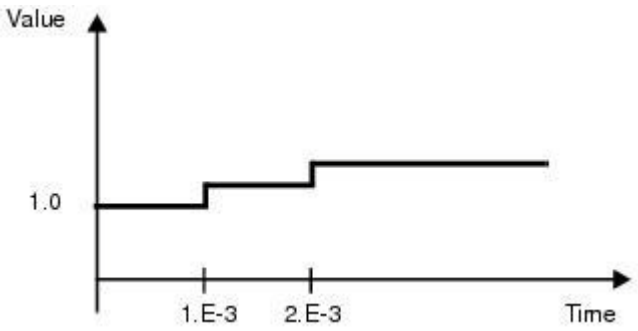
   b. DX, DY, DZ and NX, NY, NZ

→DXEL = DX/NX; DYEL = DY/NY; DZEL = DZ/NZ

2. When the starting grid point and/or element number is left blank, the default start number for the elements and grid-points is equal to the maximum number used +1. For simulations with multiple coupling surfaces, two methods of treating transport between the Euler meshes are available. One method supports meshes of TYPE=ADAPT, but does not allow the specification of starting element or starting grid-point number. This method is the default. The other method only supports meshes of TYPE=BOX but does allow for the specification of starting element number and starting grid-point number. This method is activated by PARAM, FLOW-METHOD, FACET.
3. The PID should refer to an existing property ID, which can handle the property type given by PROP.
4. To avoid that the Euler mesh is resized every time-step, the functions defined by TID-X, TID-Y, TID-Z must describe a 'step-function', as such in this example:

```
i.   TABLED1,1,,,,,,,,+1
ii.  +,0.0,1.0,,,,,,,+
iii. +,1.E-3,1.0,,,,,,,+
iv.  +.1.E-3,1.1,,,,,,,+
v.   +,2.E-3,1.1,,,,,,,+
vi.  +,2.E-3,1.2,,,,,,,+
```

which specifies the following function:



5. Care must be taken when refining the Euler mesh. To avoid instabilities, it is advised to stay within the following guidelines.
   a. For each refining step, use a scale factor larger than 0.5.
   b. Allow the solution to become smooth again after each refining step. For air bag simulations, use an interval larger than $5 * diameter_airbag/soundspeed$.
6. Resizing is not available for the Multi-material solver.
7. In most cases, METHOD=ALL is the preferred method. Using METHOD=MATERIAL may be helpful in case of instabilities due to presence of void regions.

8. A biased mesh has nonconstant element sizes in selected directories. Neighbor element size can have a constant ratio or have identical size. In literature, this type of mesh is also referred to as a nonuniform mesh or a locally refined mesh.
9. A block mesh consists of number of planes in all three directions. For a nonbiased mesh, these planes are at fixed distance from each other. In a biased mesh, the distance between subsequent planes can differ. The varying element size is determined by `1. IBIDX`. or `2. GROWX` or `3`. The constant step size specified by `X0`, `NX`, `DX`. Here, `IBIDX` overrules `GROWX` and `GROWX` overrules the `X0`, `NX`, `DX` specification. Likewise for the other directions. The locations of the planes are written out in the `OUT` file. Intersecting an x-plane with a y-plane and z-plane will give a grid point. By carrying out all intersections, the grid points are constructed.
10. `SUBMESH` glues a fine mesh into a coarse mesh and uses the same gluing functionality as PARAM, GRADED-MESH. If the fine mesh is completely contained inside the coarse mesh, no restrictions apply. Then, to avoid any restrictions, the grid points of the fine mesh are slightly displaced. But if parts of the fine mesh are outside the coarse mesh, a restriction applies. In that case, an Euler element of the coarse mesh has to be fully active or fully inactive. This means that the coarse element should not intersect elements of the fine mesh or it should be fully covered by the fine elements. Fine elements are not allowed to cover any part of the coarse elements. In practice, this means that the fine mesh has to fit nicely in the coarse mesh. For details, refer to the *Dytran User's Guide*, Chapter 2: Elements, Graded Meshes in Euler.

    When running on one cpu the elements of the Euler mesh and the submesh will be put into one euler archive. But when running with multiple cpu's the mesh and sub mesh will be put in different euler archives. To distinguish the archives the name _FVX is added to the Euler archive names. Here X is the MESH-ID MID. The Euler archive of the mesh and the submesh can be read simultaneously into Patran.
11. To determine a suitable size for the static output box, the simulation can be run first without the fields `X0BX`, `Y0BX`, `Z0BX`, `DXBX`, `DYBX`, and `DZBX` set. In the ,`OUT` file each summary of adaptive meshing gives the smallest box surrounding all adaptive elements so far. The last summary then yields the dimension of the static output box. To accommodate for elements that are not completely inside the box, the actual static output box is automatically extended a little. The actual dimensions are written in the out file after the first adaptive meshing summary. Is not needed to set the `XREF`, `YREF`, and `ZREF` options. If they are set the static output box will be compatible with the defined reference point.

12. Special care is needed for output requests of Euler elements when `TYPE=ADAPT` is used. For `SAVE` the value `1` has to be used, so each output file contains all the connectivity of the resized and/or remeshed Euler domain.
13. Defining the `CID` allows for positioning the mesh box arbitrarily in space. When active, the following restrictions apply:
    - PARAM,FASTCOUP cannot be used
    - MESH,`ADAPT` cannot be used
    - The **MESH** can only be associated with an Eulerian property. Using **MESH** with `CID`s is not supported for PSOLID
    - The use of single material Euler with strength elements is not supported.

14. Setting NELCUBE, NBX, NBY, and NBZ allows the division of the Euler mesh into a number of cubes. By setting NELCUBE equal to 2000, optimal use is made of memory caching during Euler computation. This can give a speedup of 1.5. Defining NBX overrules the definition of NELCUBE. If NBX is defined, NBY and NBZ also need to be defined.

    When using multiple Euler cubes, the BARRIER and FLOW definitions only support geometric conditions like boundary face direction and square definition.

15. There are several ways to distribute cubes across processors. Some ways may lead to bad load balancing. To avoid this, it is possible to control the way Euler cubes are distributed across processors by defining PROCDIR.

    When the option PROCDIR=USER, the values for NBX, NBY, and NBZ must be such that NBX is equal or a multiple of NPX,NBY is equal or a multiple of NPY, and NBZ is equal or a multiple of NPZ. Also, for this option, NPX*NPY*NPZ must be equal to the number of processors used in the cluster.

    For the option PROCDIR=SIMPLE, the values NBX, NBY, and NBZ on PARAM EULERCUB must be such and NBX*NBY*NBZ is equal or a multiple of the number of processors used. For instance, if the number of processors in the cluster if 4, NBX*NBY*NBZ must be equal to either 4, 8, or 12, etc. Otherwise, the calculation will terminate prematurely with an error message.

    When using the options, the BARRIER and FLOW definitions only support geometric conditions like boundary face direction and a square definition.

MOMENT

Concentrated Moment or Enforced Motion

This entry is used in conjunction with a TLOADn entry and defines the location where the moment or enforced motion acts as well as the direction and scale factor.

Format and Example

1	2	3	4	5	6	7	8	9	10
MOMENT	LID	G		SCALE	N1	N2	N3		
MOMENT	2	5		2.9		1.0			

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid-point number or rigid body where the load is applied	See Remark 5.	Required
SCALE	Scale factor for the moment	R	1.0
N1, N2, N3	Components of a vector giving the direction of the moment. At least one must be nonzero.	R	See Remark 4.

Remarks

1. At time t, the moment $M(t)$ is given by $M(t) = SCALE*N*T(t)$

   where $SCALE$ is the scale factor; N is the vector given by N1, N2, and N3; and $T(t)$ is the value at time t interpolated from the table referenced on the TLOADn entry.

2. Moments can also be defined on the DAREA entry.
3. LID must be referenced by a TLOADn entry.
4. If a component field N1, N2, and/or N3 is left blank:

   Moment prescription – The component of the moment is equal to zero.

   Velocity prescription – The component of the angular velocity is not restrained.

5. If G references a MATRIG, an RBE2-FULLRIG, or a RIGID surface, the load is applied to the center of the rigid body. If G references a MATRIG, G must be MR<id>, where id is the MATRIG number. If G references a RBE2-FULLRIG, G must be FR<id>, where id is the RBE2 number. If G references a RIGID surface, G is the RIGID surface number.
6. If the TYPE field on the TLOAD entry is 0, this defines a moment applied to a grid point. If the TYPE field is 2, it defines an enforced motion on the grid point. If the TYPE field is 12, it defines an enforced motion applied to the center of the rigid body, and if the TYPE field is 13, it defines a moment applied to the center of a rigid body.

MOMENT1

Follower Moment, Form 1

This entry is used in conjunction with a TLOADn entry and defines a follower moment with direction that is determined by two grid points. MOMENT1 can be applied to any type of grid point.

Format and Example

1	2	3	4	5	6	7	8	9	10
MOMENT1	LID	G	SCALE	G1	G2				
MOMENT1	2	5	2.9	16	13				

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid-point number where the moment is applied	I > 0	Required
SCALE	Scale factor for the moment	R	1.0
G1, G2	Grid-point numbers. The direction of the moment is a vector from G1 to G2. G1 must not be the same as G2.	I > 0	Required

Remarks

1. At time t, the moment $\underline{M}(t)$ is given by

   $(t) = SCALE*N*T($

   where $SCALE$ is the scale factor, $\underline{N}$ is the vector from G1 to G2, and $T(t)$ is the value at time t interpolated from the table referenced by the TLOADn entry.
2. LID must be referenced by a TLOADn entry.
3. The MOMENT1 entry defines a follower moment in that the direction of the moment changes as the grid points G1 and G2 move during the analysis.

MOMENT2

Follower Moment, Form 2

This entry is used in conjunction with a TLOADn entry and defines a follower moment with direction that is determined by four grid points. MOMENT2 can be applied to any type of grid point.

Format and Example

1	2	3	4	5	6	7	8	9	10
MOMENT2	LID	G	SCALE	G1	G2	G3	G4		
MOMENT2	2	5	2.9	16	13	17	18		

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid-point number where the moment is applied	I > 0	Required
SCALE	Scale factor for the moment	R	1.0
G1-G4	Grid-point numbers. The moment direction is determined by a vector product of the vectors G1 to G2 and G3 to G4. (G1 must not be the same as G2, and G3 must not be the same as G4.)	I > 0	Required

Remarks

1. At time t, the moment $\underline{M}(t)$ is given by:

   $(t) = SCALE*\underline{N}*T($

   where $SCALE$ is the scale factor, $\underline{N}$ is the vector product of the vectors from G1 to G2 and G3 to G4. respectively, and $T(t)$ is the value at time t interpolated from the table referenced by the TLOADn entry.
2. LID must be referenced by a TLOADn entry.
3. The MOMENT2 entry defines a follower moment in that the direction of the moment changes as the grid points G1, G2, G3, and G4 move during the analysis.

NASINIT

MSC Nastran Initialization

Definition of the logistics of a Dytran prestress run.

Format and Example

1	2	3	4	5	6	7	8	9	10
NASINIT	STEPS	DAMP	TNOACT	FACTOR					
NASINIT	1000	YES	1.E-02	0.01					

Field	Contents	Type	Default
STEPS	Number of steps used to set the grid point displacement.	I > 0	1
DAMP	Request for additional relaxation phase after displacement phase (Yes/No).	C	No
TNOACT	End time of relaxation phase.	R > 0.0	1.E20
FACTOR	Viscous-damping factor.	R > 0.0	0.001

Remarks

1. The time step is constant during the displacement phase and is defined by PARAM,INISTEP.
2. Damping is optional and is not always necessary.
3. The deformed geometry grid-point data is written out after the displacement phase, if no damping is requested, or after the relaxation phase, when the DAMP field is set to YES. The same applies to the solution file. (See also the SOLUOUT and BULKOUT FMS statements.)
4. The displacements from an MSC Nastran solution are imposed by an enforced velocity field calculated from the displacements and control parameters.
5. All boundary conditions and loads defined are deactivated during the displacement phase and are activated after the displacement phase ends.
6. Note that although the deformed geometry after the displacement phase is exactly the same as the MSC Nastran geometry, the actual stress state may differ due to differences in Dytran and MSC Nastran element formulations.
7. Lagrangian CHEXA, CQUAD4, and CTRIA3 elements can be initialized, but the shell membranes cannot.
8. The viscous-damping factor will overwrite the values provided by VISCDMP entry for Lagrangian CHEXA, CQUAD4, and CTRIA3 elements, except when viscous-damping is defined using TABLED1 in VISCDMP entry.
9. For prestressing rotating structures, it is recommended that a centrifugal force field be used, rather than a rotational velocity field. In the actual transient dynamic analysis, the centrifugal force field can be replaced by a rotational velocity field with consistent boundary conditions.
10. Make the problem setup for the final transient analysis consistent with the prestress analysis.

PARAM

Parameter

Defines the values for parameters used during the solution.

Format and Example

1	2	3	4	5	6	7	8	9	10
PARAM	NAME	V1							
PARAM	REZTOL	0.1							

Field	Contents	Type	Default
NAME	Parameter name	C	Required
V1	Value associated with NAME	I, R, C	See Chapter 6: Parameters.

Remarks

1. A list of the parameters that can be set, along with the parameter names and values, is given in Chapter 6: Parameters.
2. PARAM entries do not necessarily have to be located in the Bulk Data Section.
3. PARAM values can be redefined during restarts.

PBAR

Simple Beam Property

Defines the properties of a simple beam (bar) that is used to create bar elements via the CBAR entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
PBAR	PID	MID	A	I1	I2	J			
PBAR	39	6	2.9		5.97				

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
MID	Material number	I > 0	Required
A	Area of bar cross section	R > 0.0	Required
I1, I2	Area moments of inertia	R ≥ 0	Required
J	Torsional constant	R ≥ 0	0.0

Remarks

1. I1 is the moment of inertia about the element z-axis, I_{zz}.

   I2 is the moment of inertia about the element y-axis, I_{yy}.

2. This element is solved as a Belytschko-Schwer beam.

PBCOMP

Beam Alternate Form of PBEAM

Alternate form of the PBEAM entry to define properties of a uniform cross-sectional beam referenced by a CBEAM entry. This entry is also used to specify lumped areas of the beam cross section for nonlinear analysis and/or composite analysis.

Format and Example

1	2	3	4	5	6	7	8	9	10
PBCOMP	PID	MID	A						+
PBCOMP	181	6	2.9						+
+	SHFACT				N1	N2	SYMOPT		+
+							1		+
+	Y1	Z1	C1	MID1					+
+	-0.5	1.2	0.1	18					+
+	Y2	Z2	C2	MID2					+
+	0.2	0.9	0.15						+
+	Yi	Zi	Ci	MIDi					
+	...	...	...						

Field	Contents	Type	Default
PID	Unique property number	I ≥ 0	Required
MID	Material identification number	I ≥ 0	Required See Remark 2.
A	Area of beam cross section	R> 0.0	Required
SHFACT	Shear factor for the section	R	0.83333
N1, N2	The (y,z) coordinates of neutral axis. See the figure in the CBEAM entry description.	R	0.0
SYMOPT	Symmetry option to input lumped areas for the beam cross section. See Figure 5-12.	1≤ I≤5	Required
Yi, Zi	The (y,z) coordinates of the lumped areas in the element coordinate system.	R	0.0 See Remark 1.
Ci	Fraction of the total area for the i^{th} lumped area	R > 0.0	Required
MIDi	Material identification number for the i^{th} integration point	I > 0	MID

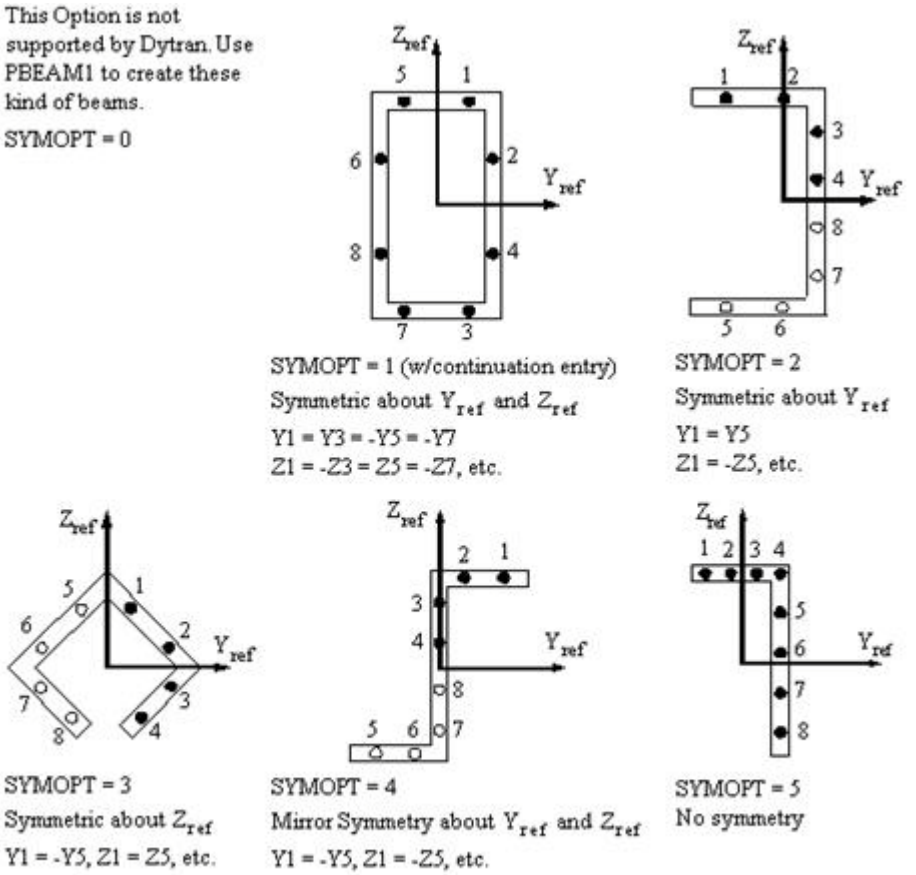


Figure 5-12 PBCOMP Entry SYMOPT Type Examples with Eight Lumped Areas.

Remarks

1. The `PID` number must be unique with respect to other `PBCOMP` entries as well as PBEAM entries. The second continuation entry may be repeated 18 more times. If SECTION = 5 a maximum of 21 continuation entries is allowed; that is, a maximum of 20 lumped areas may be input. If SECTION=1 through 4, the total number of areas input plus the total number generated by symmetry must not exceed 20.
2. If the `MIDi` field on a continuation entry is blank, the value will be that of `MID` on the parent entry. `MIDi` values may be input on continuations without the corresponding `Yi`, `Zi`, and `Ci` values to allow different stress-strain laws.

PBEAM

Beam Property

Defines the properties of the CBAR and CBEAM element.

Format and Example

1	2	3	4	5	6	7	8	9	10
PBEAM	PID	MID	A(A)	I1(A)	I2(A)		J(A)		+
PBEAM	7	14	3.6	24.9	24.9		22.6		+
+									+
+									+
+		X/XB	A(B)	I1(B)	I2(B)		J(B)		
+		1.0	3.6	24.9	24.9		22.6		

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
MID	Material number	I > 0	PID
A(A)	Area of the beam cross section at end A of the beam	R > 0.0	Required
I1(A)	Area moment of inertia about the beam-element's z-axis at end A of the beam	R > 0.0.	Required
I2(A)	Area moment of inertia about the beam-element's y-axis at end A of the beam	R > 0.0.	Required
J(A)	Torsion constant at end A of the beam	R	0.0
X/XB	For MSC Nastran, this is the distance along the beam from end A divided by the length of the beam. The properties are defined at several positions along the beam's length. For Dytran, all the intermediate positions are ignored. The only relevant data occurs when X/XB is 1.0. corresponding to end B of the beam.	R	Required
A(B)	Area of the cross section at end B of the beam	R > 0.0	Required
I1(B)	Area moment of inertia about the beam-element's z-axis at end B of the beam	R > 0.0	Required
I2(B)	Area moment of inertia about the beam-element's y-axis at end B of the beam	R > 0.0	Required
J(B)	Torsion constant at end B of the beam	R	0.0

Remarks

1. This entry is an alternative to thePBARentry and defines exactly the same element and properties. It is more complicated to use than PBAR and has no advantages. PBEAM is retained for compatibility with MSC Nastran and those modeling packages that write PBEAM entries. Use the PBAR entry if you can.
2. A Belytschko-Schwer beam is used with a shear factor of 0.83333. The plastic moduli are assumed to be those for a rectangular section

   $$Zp_y = \sqrt{0.75*A*I2}$$

   $$Zp_z = \sqrt{0.75*A*I1}$$

   To specify values of Z_p for other sections, use the PBEAM1 entry.
3. For more complex beam properties, use the PBEAM1 entry.
4. Note the following:

   $$I_1 = I_{zz} \quad I_2 = I_{yy} \quad J = I_{xx}$$

PBEAM1 Beam Properties (Belytschko-Schwer)

Defines complex beam properties that cannot be defined using the PBAR or PBEAM entries. These entries are to be used only for Belytschko-Schwer elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
PBEAM1	PID	MID	FORM			SHFACT	SECT		+
PBEAM1	1	7	BELY			0.9	RECT		+
+	A	I1	I2	J	ZPZ	ZPY			+
+									+
+	CS1	CS2	CS3	CS4	CS5	CS6			
+									

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
MID	Material number	I > 0	PID
FORM	Element formulation:	C	Required
	BELY Belytschko-Schwer		
SHFACT	Shear factor for the section.	R	0.83333
SECT	Type of section. See Remark Step 4	C	RECT
A	Area of the section	R	Blank
I1	The moment of inertia about the element z-axis	R	Blank
I2	The moment of inertia about the element y-axis	R > 0.0	Blank
J	The torsional stiffness of the section	R ≥ 0	Blank
ZPZ	Plastic modulus *Zp* about the element z-axis	R > 0.0	Blank
ZPY	Plastic modulus *Zp* about the element y-axis	R > 0.0	Blank
CSi	Geometrical definition of the cross section. The data in these fields depends on the type of the section.	R ≥ 0.	See Step 4

Remarks

1. Only the entries that are relevant for Belytschko-Schwer beam definition are listed. PBEAM1 entries that apply to Hughes-Liu beams appear earlier in this PBEAM1 discussion.
2. Note the following:

$$I_1 = I_{zz} \quad I_2 = I_{yy} \quad J = I_{xx}$$

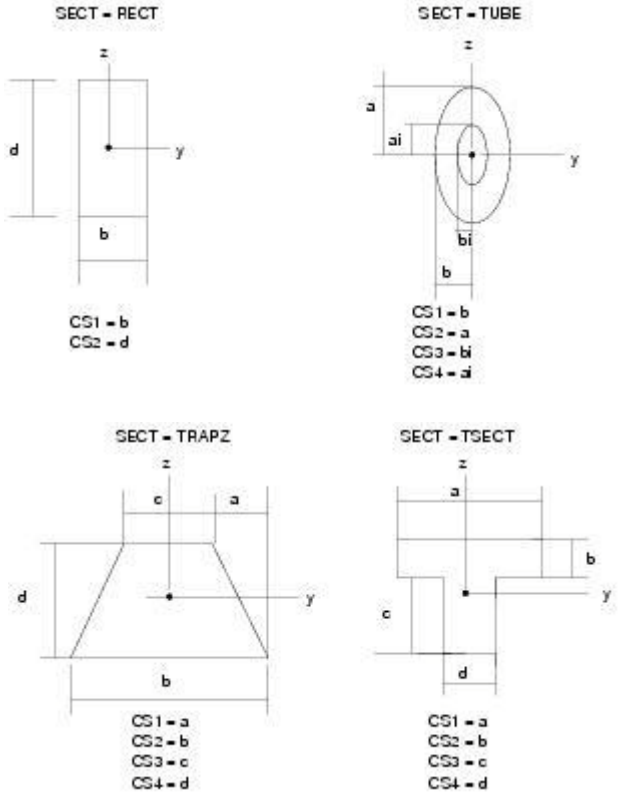
3. The cross-sectional properties are calculated as follows:

Step 1:If the geometry is defined in the fields CSi, the values of A, I1, I2, J, ZPZ, and ZPY are automatically calculated.

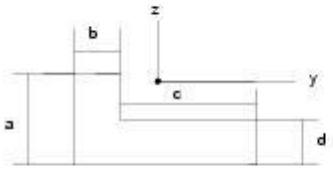
Step 2:If a value is defined in the fields A, I1, I2, J, ZPZ, and ZPY, these values override the values as calculated in Step 1.

Step 3:All values of CSi for a particular cross section (see Remark Step 4) must be entered for the geometry to be defined. If not all values of CSi are supplied, then values for A, I1, I2 and J are required, and ZPZ and ZPY have a default value of 1E20.

Step 4:The geometrical definitions for the various cross sections are defined in the element coordinate system as follows:

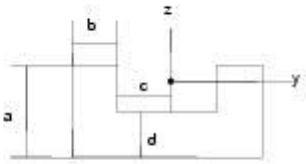


SECT = LSECT



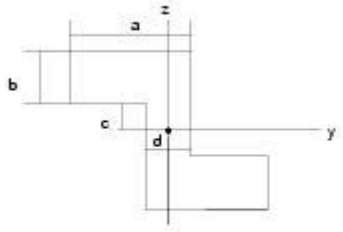
CS1 = a
CS2 = b
CS3 = c
CS4 = d

SECT = USECT



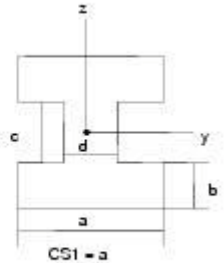
CS1 = a
CS2 = b
CS3 = c
CS4 = d

SECT = ZSECT



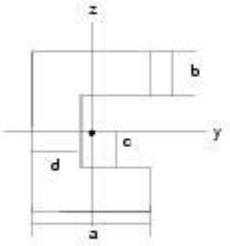
CS1 = a
CS2 = b
CS3 = c
CS4 = d

SECT = ISECT



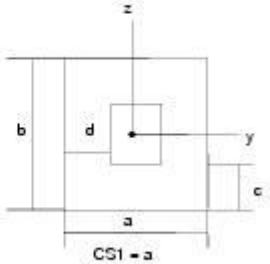
CS1 = a
CS2 = b
CS3 = c
CS4 = d

SECT = CSECT



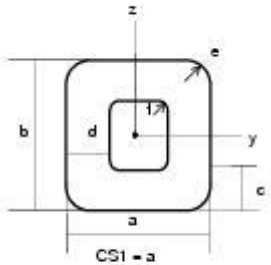
CS1 = a
CS2 = b
CS3 = c
CS4 = d

SECT = BOXSECT



CS1 = a
CS2 = b
CS3 = c
CS4 = d

SECT = RCBSECT



CS1 = a
CS2 = b
CS3 = c
CS4 = d
CS5 = e
CS6 = f

PBEAM1

Beam Properties (Hughes-Liu)

Defines more complex beam properties that cannot be defined using the PBAR or PBEAMentries. The following entries are for Hughes-Liu beam elements only.

Format and Example

1	2	3	4	5	6	7	8	9	10
PBEAM1	PID	MID	FORM	QUAD	NUMB	SHFACT	SECT		+
PBEAM1	1	7	HUGHES	GAUSS		0.9	RSECT		+
+	V1	V2	V3	V4					
+	30.1	30.1	10.0	10.0					

Field	Contents		Type	Default
PID	Unique property number		I > 0	Required
MID	Material number		I > 0	PID
FORM	Element formulation		C	Required
	HUGHES	Hughes-Liu		
QUAD	Type of quadrature:		C	GAUSS
	GAUSS	Gauss quadrature		
	LOBATTO	Lobatto quadrature		
NUMB	The number of integration points for Hughes-Lui beams. For Gauss integration, the following can be specified:		1 > 0	3
	1	1 point (rod element)		
	2	2 x 2 points (4-point circle, if tubular)		
	3	3 x 3 points (9-point circle, if tubular)		
	4	4 x 4 points (16-point circle, if tubular)		
	At present only 3 x 3 points are available with the Lobatto quadrature.			
SHFACT	Shear factor for the section		R	0.83333
SECT	Type of section:		C	RECT
	RECT	Rectangular cross section		
	TUBE	Tubular cross section		
V1-V4	Geometric properties of the beam. The data in these fields depends on the beam formulation and the type of cross section.			Required
	For Hughes formulations with rectangular cross sections:			
	V1	The thickness in the element y direction at grid point 1		

Field	Contents		Type	Default
	V2	The thickness in the element y direction at grid point 2		
	V3	The thickness in the element z direction at grid point 1		
	V4	The thickness in the element z direction at grid point 2		
	For Hughes formulations with tubular cross sections:			
	V1	The outer diameter at grid point 1		
	V2	The outer diameter at grid point 2		
	V3	The inner diameter at grid point 1		
	V4	The inner diameter at grid point 2		

Remarks

1. Only the entries that are relevant for the Hughes-Liu beam definition are listed here. PBEAM1 entries that apply to Belytschko-Schwer beams are provided in the previous PBEAM1 description. For more complex cross-sections with the Hughes-Liu beam definition, please use the Predefined Hughes Liu Section (HLSECTS) option.

PBEAM1 Beam Properties (Predefined Hughes-Liu Cross Sections)

Defines more complex beam properties that cannot be defined using the PBAR or PBEAM entries. The following entries are for predefined cross sections of Hughes-Liu beam elements only.

Format and Example

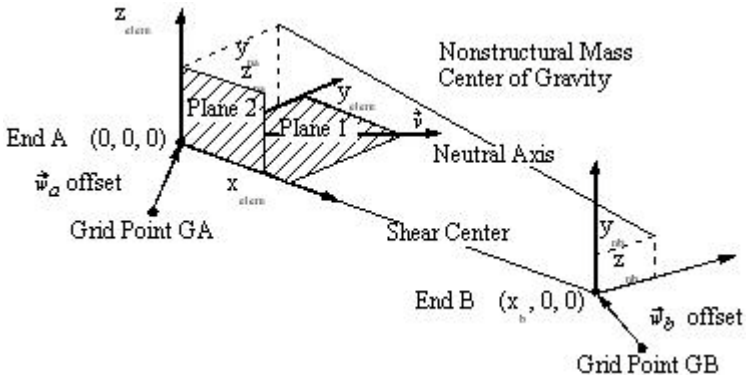
1	2	3	4	5	6	7	8	9	10
PBEAM1	PID	MID	FORM	DATABASE		SHFACT	SECT		+
PBEAM1	1	7	HLSECTS	DYTRAN		0.9	ISECT		+
+	V1	V2	V3	V4	V5	V6	V7	V8	+
+	30.1	30.1	10.0	10.0					+
+	N1(A)	N2(A)	N1(B)	N2(B)					+
+	2.0		2.0						+
+	V9	V10	V11	V12					+
+									+
+	N1(A)	N2(A)	N1(B)	N2(B)					
+	2.0		2.0						

Field	Contents		Type	Default
PID	Unique property number		1 > 0	Required
MID	Material number		1 > 0	PID
FORM	Element formulation		C	Required
	HLSECTS	Predefined Hughes-Liu cross sections		
DATABASE	Cross-section database		C	Required
	DYTRAN	See figures in Remark 3. for available cross sections.		
	NASTRAN	See figures in Remark 4. for available cross sections.		
SHFACT	Shear factor for the section		R	0.83333
SECT	Type of section.		C	Required
	ZSECT	Z cross-section		
	LSECT	L cross-section		
	TSECT	T cross-section		

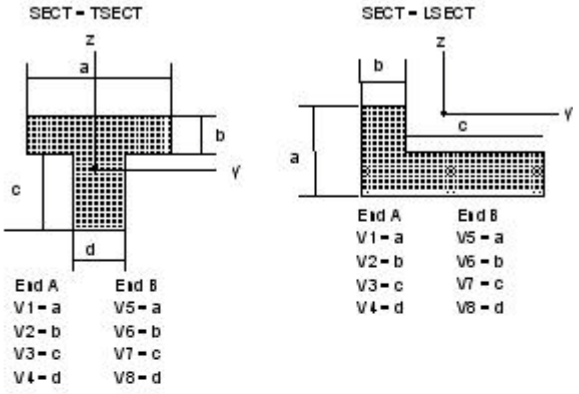
Field	Contents		Type	Default
	USECT	U/CHAN2 cross-section		
	ISECT	I cross section		
	BOXSECT	BOX cross-section. (MSC Nastran Database only)		
	HATSECT	HAT cross-section. (MSC Nastran Database only)		
	RCBSECT	“Round Corners BOX” cross-section (MSC Nastran Database only).		
V1 - V4	Geometric properties of the beam. The data in these fields depends on the beam formulation and the type of cross section.			
	For Hughes formulations of the Dytran database cross sections.			
	V1-V4	Cross Section Dimensions at end A	R	Required
	V5-V8	Cross Section Dimensions at end B	R	Same as V1-V4
	For Hughes formulations of the MSC Nastran Database cross sections:			
	V1-V6	Cross Section Dimensions of beam	R	Required
N1(A), N2(A), N1(B), N2(B)	(y,z) coordinates of neutral axis for end A and end B. See the figure in Remark 2.		R	0.0

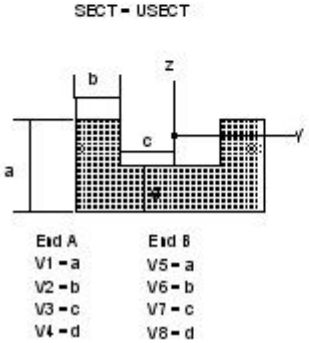
Remarks

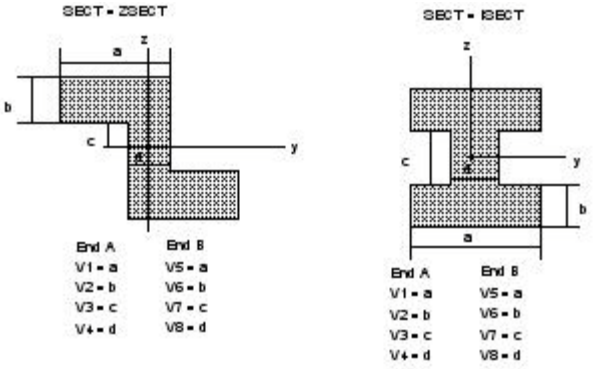
1. Only the entries that are relevant for the predefined Hughes-Liu beam definition are listed here. PBEAM1 entries that apply to Belytschko-Schwer beams are provided in PBEAM1 (Belytschko-Schwer).
2. The cross sections TUBE and RECT can be defined in the regular Hughes-Liu PBEAM1 entry.



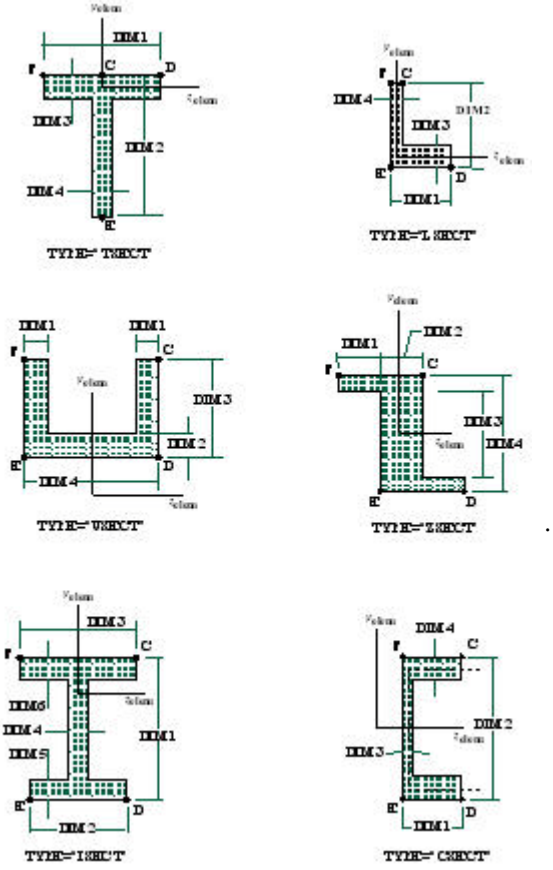
3. The following cross sections can be defined using the Dytran database.

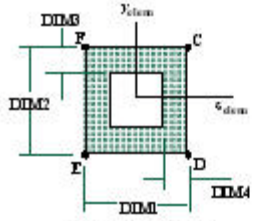




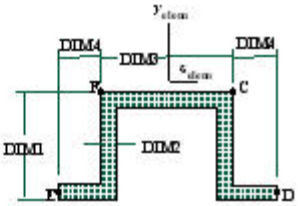


4. The following figures are part of the MSC Nastran cross-sections database

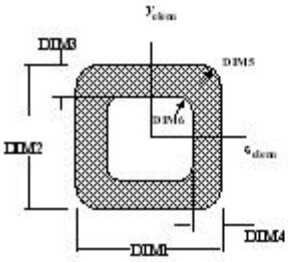




TYPE='BOXSECT'



TYPE='HATSECT'



TYPE='RCBSECT'

PBEAML

Beam Cross-Section Properties

Defines the properties of the CBARand CBEAM element by cross-sectional dimensions.

Format and Example

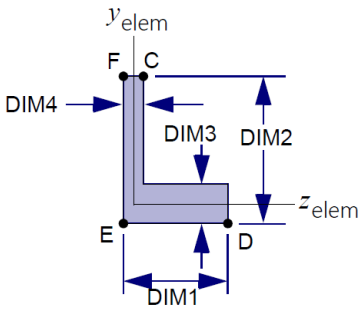
1	2	3	4	5	6	7	8	9	10
PBEAML	PID	MID		TYPE					+
PBEAML	7	14		HAT					+
+	DIM1	DIM2	DIM3	DIM4	DIM5	DIM6			
+	.8	.1	.4	.3					

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
MID	Material number	I > 0	PID
TYPE	Cross-section shape. See Remark 4. (TUBE, L, I, CHAN, T, BOX, BAR, Z, CHAN2, HAT, RCBOX)	C	Required
DIMi	Cross-section dimensions	R > 0.0	Required

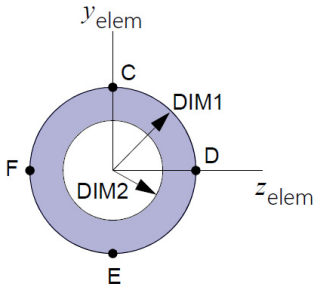
Remarks

1. For structural problems, PBEAML entries must reference a MAT1 orDMATEP material entry.
2. The property number PID must be unique with respect to all other PBEAMand PBEAML property numbers.
3. The PBEAML entry is automatically translated into a Hughes-Liu PBEAMentry.
4. See the PBEAM1entry description for a discussion of beam-element geometry. The BAR and TUBE sections are translated into Hughes-Liu PBEAM1definition with the Gaussian quadrature. The others are divided into a predefined pattern of integration points.

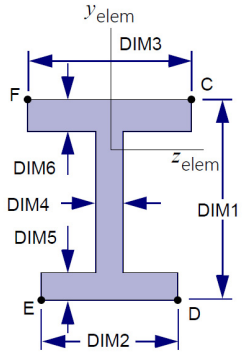
5. Following is an overview of the available cross sections and the specific definitions valid for these cross sections.



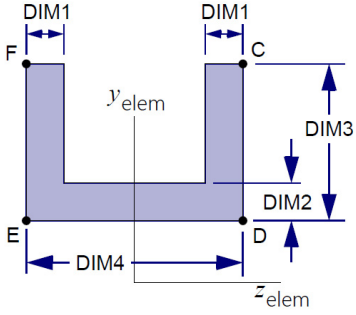
TYPE="L"



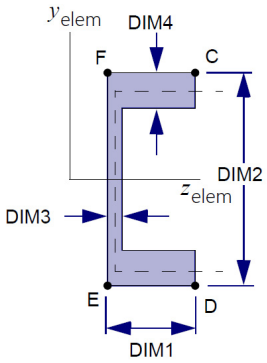
TYPE="TUBE"



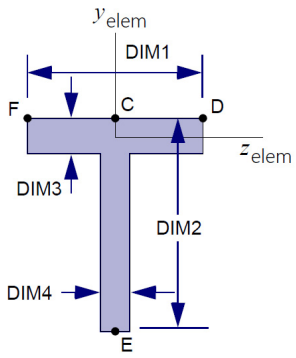
TYPE="I"



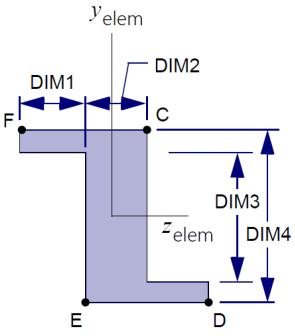
TYPE="CHAN2"



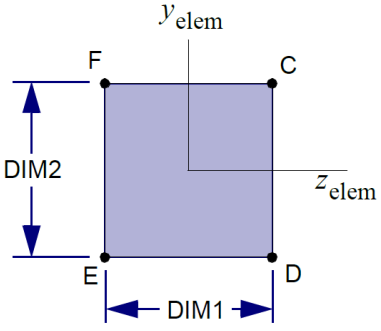
TYPE="CHAN"



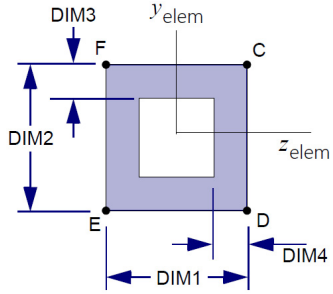
TYPE="T"



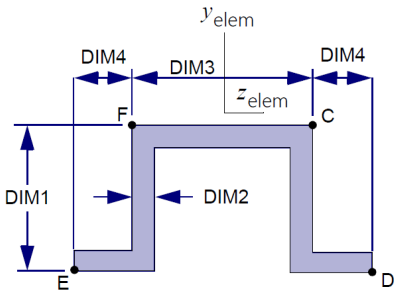
TYPE="Z"



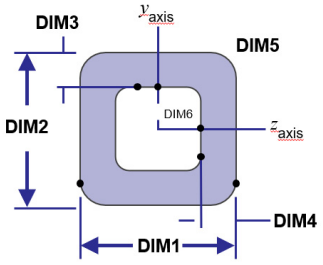
TYPE="BAR"



TYPE="BOX"
(Warping computed as zero if MDLPRM,TWBRBML,1)



TYPE="HAT"



TYPE="RCBOX"

PBELT

Belt Property

Defines the properties of a belt element referenced by a CROD entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
PBELT	PID	LOAD	UNLOAD	DENSITY	DAMP1	DAMP2	SLACK	PRESTRESS	
PBELT	9	12	12	2.E-5	0.1	0.1			

Field	Contents	Type	Default
PID	Unique belt property number	I > 0	Required
LOAD	Number of a TABLED1 defining the force as a function of strain during loading. The strain at time n is specified as engineering strain:	I > 0	Required
	$\iota in(n) = \frac{(length(n) - length}{(length(0))}$		
UNLOAD	Number of a TABLED1 defining the force as a function of strain during unloading. The strain at time n is specified as engineering strain:	I > 0	Required
	$in(n) = \frac{(length(n) - length(}{(length(0))}$		
DENSITY	Density of the belt elements as mass per unit length	R > 0.0	Required
DAMP1	A damping force is added to the internal force of the belt elements to damp out high frequency oscillations. The damping force is equal to:	R > 0.0	0.1
	$F_{damp} = DAMP1*(mass)*(dvel)/(dt)$		
	where		
	F_{damp} = damping force		
	$DAMP1$ = damping coefficient		
	$mass$ = mass of belt element		
	$dvel$ = velocity of elongation		
	dt = time step		
DAMP2	The damping force is limited to:	R > 0.0	0.1
	$DAMP2*F_{belt}$		
	where		
	F_{belt} = is the internal force in the belt element.		
SLACK	Number of a TABLED1 defining the slack as a function of time. The slack must be specified as engineering strain and will be subtracted from the element strain at time n as:	I > 0	Blank
	$strain(n) = strain(n) - SLACK(n)$		

Field	Contents	Type	Default
	The force in the element is zero until the element strain exceeds the slack.		
PRESTRESS	Number of a TABLED1 defining a prestress strain as a function of time. The prestress strain must be specified as engineering strain and will be added to the element strain at time n as:	I > 0	Blank
	$strain(n) = strain(n) + PRESSTRESS(n)$		

Remarks

1. The loading and unloading curves must start at (0.0, 0.0).
2. During loading, the loading curve is applied to determine the force in the belt element. At unloading, the unloading curve is shifted along the strain axis until it intersects the loading curve at the point from which unloading commences. The unloading table is applied for unloading and reloading, until the strain again exceeds the intersection point. Upon further loading, the loading table is applied. For subsequent unloading, the sequence is repeated.
3. Belt elements are tension only elements.
4. Instantaneous slack of an element can also be initialized per element using the TICEL entry with the keyword SLACK and a corresponding VALUE.

PBUSH — Generalized Spring-and-Damper Property

Defines the nominal property values for a generalized spring-and-damper structural element.

Format and Example

PBUSH	PID	"K"	K1	K2	K3	K4	K5	K6	+
PBUSH	3	K	4.35	5.3	7.1	0.5	0.4	0.1	+
+		"B"	B1	B2	B3	B4	B5	B6	+
+		B	0.06	0.07	0.08				+
+		"M"	M						
+		M	0.1						

Field	Contents	Type	Default
PID	Property identification number.	I > 0	Required
"K"	Flag indicating that the next 1 to 6 fields are nominal stiffness values in the element coordinate system.	C	
Ki	Nominal stiffness values in directions 1 through 6.	R ≥ 0.0	0.0
"B"	Flag indicating that the next 1 to 6 fields are force-per-velocity damping coefficients.	C	
Bi	Nominal damping coefficients in direction 1 through 6 in the element coordinate system.	R ≥ 0.0	0.0
"M"	Flag indicating that the following entries are mass properties for the Bush element.	C	
M	Lumped mass of the Bush. See Remark 2.	R ≥ 0.0	0.0

Remarks:

1. All components are based on the local coordinate system.
2. The lumped mass defined in PBUSH will be distributed to grid points according to S values in the CBUSH entry.
3. Unlike PBUSH implementation in Nastran, the structural damping fields, stress or strain coefficients fields, and thermal properties are not supported.

PBUSHT Nonlinear Force Deflection Spring Property

Defines the stress-dependent properties for a generalized spring-and-damper element.

Format and Example

PBUSHT	PID	KN	TKNID1	TKNID2	TKNID3	TKNID 4	TKNID 5	TKNID 6	+
PBUSHT	3	KN	101	102	103	104	105	106	+
+			FDC	FUSE	DIR	OPTION	LOWER	UPPER	+
+			TRYZ	1	23	ULTLD	-1000.	2000.	+
+			FDC	FUSE	DIR	OPTION	LOWER	UPPER	
+			NR	1	1		-5000.	8000.	

Field	Contents	Type	Default
PID	Property identification number that matches the identification number on a PBUSH entry.	I > 0	Required
KN	Flag indicating that the next 1 to 6 fields are nonlinear force-deflection table identification numbers.	C	
TKNIDi	Identification number of a TABLEDi entry that defines the force vs. deflection relationship in directions 1 through 6.	I > 0	

Field	Contents	Type	Default
FDC	Force deflection curve rule. Specifies a dependence between displacement components. NR: Implies there is no force-deflection rule. Translational and rotational DOFs are all independent of each other. TRXY: A radial dependence exists between u_x and u_y element displacements. Either K1 or K2 may be specified on the PBUSHentry. If both are specified, then K1 will be used. On the PBUSHT KN entry, either TKNID1 or TKNID2 may be specified. If both are specified, then TKNID1 is used. The expression $u_r = \sqrt{(u_x)^2 + (u_y)^2}$ is used for table lookup. Under the OPTION=ULTLD, the force used for the fuse calculation is computed using u_r. Under the OPTION=RULTLD, the true forces f_x and f_y in the springs "x" and "y" are used for the fuse calculation. New K1 is returned from the slope of the load-deflection curve at the slope at current u_r. TRXZ: A radial dependence exists between u_x and u_z element displacements. Either K1 or K3 may be specified on the PBUSH entry. If both are specified, then K1 will be used. On the PBUSHT KN entry, either TKNID1 or TKNID3 may be specified. If both are specified, then TKNID1 is used. The expression $u_r = \sqrt{(u_x)^2 + (u_z)^2}$ is used for table lookup. Under the OPTION=ULTLD, the force used for the fuse calculation is computed using u_r. Under the OPTION=RULTLD, the true forces f_x and f_z in the springs "x" and "z" are used for the fuse calculation. New K1 is returned from the slope of load-deflection curve at slope at current u_r.	C	NR

Field	Contents	Type	Default
	TRYZ: A radial dependence exists between u_y and u_z element displacements. Either K2 or K3 may be specified on the PBUSH entry. If both are specified, then K2 will be used. On the PBUSHT KN entry, either TKNID2 or TKNID3 may be specified. If both are specified, then TKNID2 is used. The expression $u_r = \sqrt{(u_y)^2 + (u_z)^2}$ is used for table lookup. Under the OPTION=ULTLD, the force used for the fuse calculation is computed using u_r. Under the OPTION=RULTLD, the true forces f_y and f_z in the springs "y" and "z" are used for the fuse calculation. New K2 is returned from slope of load-deflection curve at slope at current u_r. TS: A spherical dependence exists between ux, uy, and uz element displacements. Either K1, K2, or K3 may be specified on the PBUSH entry. If all three are specified K1 is used. On the PBUSHT KN entry either TKNID1, TKNID2, or TKNID3 may be specified. If all three are specified, then TKNID1 is used. If only TKNID2 and TKNID3 are specified, then TKNID2 is used. The expression $u_s = \sqrt{(u_x)^2 + (u_y)^2 + (u_z)^2}$ is used for table lookup. Under the OPTION=ULTLD, the force used for the fuse calculation is computed using us. Under the OPTION=RULTLD, the true forces f_x, f_y, and f_z in the springs "x", "y", and "z" are used for the fuse calculation. New K1 is returned from slope of load-deflection curve at slope at current u_s.		
FUSE	0: The element remains active irrespective of failure level. 1: The element is deactivated if maximum failure as specified in OPTION is reached.	I = 0 or 1	0
DIR	The fuse direction component numbers. Any unique combination of the integers 1 through 6 with no embedded blanks. Eq.0 any direction is allowed to fuse. Same if user entered 123456.	C	0

Field	Contents	Type	Default
OPTION	Specifies a failure mode. ULTLD: The specified failure load in compression or tension will be used to define failure using load computed from the appropriate radial dependence of displacement relationship. RELDIS: The specified max relative positive displacement will be used to define failure. RULTLD: The specified failure load in compression or tension will be used to define failure using the actual current physical load in each spring.	C	RELDIS
LOWER	Lower failure bound. If OPTION = ULTLD then LOWER specifies a lower failure load If OPTION = RELDIS then LOWER specifies a minimum relative displacement before failure.	R ≤ 0.0	-1.0E20
UPPER	Upper failure bound. If OPTION = ULTLD then UPPER specifies an upper failure load. If OPTION = RELDIS then UPPER specifies a maximum relative displacement before failure.	R ≥ 0.0	1.0E20

Remarks:

1. For nonlinear analysis the nominal values are used at the beginning of the analysis to compute the initial loading of the CBUSH element and meaningful values should be supplied for stiffness. As the nonlinear analysis proceeds, the user supplied load deflection curves are used to compute new stiffness values.
2. If FUSE=0, the remaining entries are ignored. If FUSE > 0, the OPTION, UPPER, and LOWER fields must be specified.
3. The following relationships exists between FDC and DIR

FDC = TRXY	1 or 2 or 12 in the DIR field applies to the radial rule	a 3 in the DIR field will be ignored.
FDC = TRXZ	1 or 3 or 13 in the DIR field applies to the radial rule	a 2 in the DIR field will be ignored.

FDC = TRYZ	2 or 3 or 23 in the DIR field applies to the radial rule	a 1 in the DIR field will be ignored.
FDC = TS	Any unique combination of the integers 1 through 3 with no embedded blanks in the DIR field applies to the spherical rule	

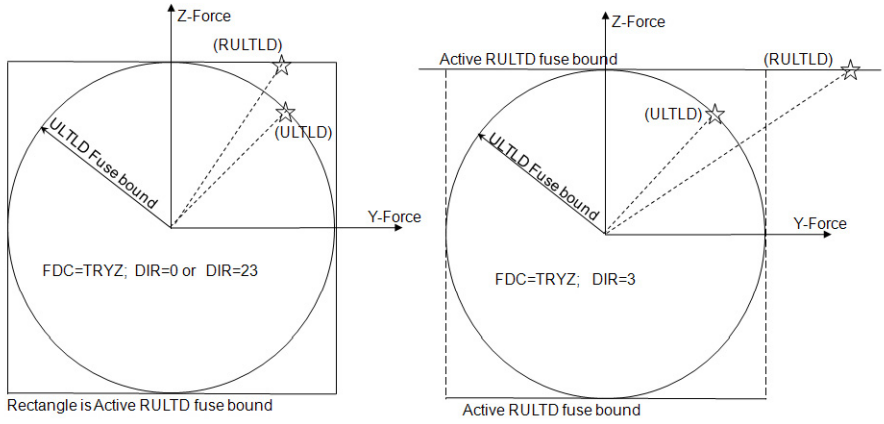
4. The figures below depict, using FDC=TRYZ as an example, the OPTION=ULTLD, and OPTION=RULTLD. Under the OPTION=ULTLD the resultant F_r load when FUSED will be

   $$F_r = \sqrt{(F_y)^2 + (F_z)^2} \approx Limit.$$

   Under the OPTION=RULTLD the resultant Fr load when FUSED will in general be

   $F_r = \sqrt{(F_y)^2 + (F_z)^2} > Limit$ with one of the Fy or Fz ≈ LIMIT. For FDC=TRYZ and DIR=123, the T1 direction is not included in the force direction curve rule, thus possible fusing in the T1 direction is always based on the actual T1 spring displacement and Fx if active will be Fx ≈ LIMIT. For FDC=NR fusing is always calculated based on each individual spring displacement and the actual individual spring loading.

   .



5. CBUSH element force is equal to the summation of defined spring and damper. However, the failure of CBUSH element is assessed merely based on the spring load.
6. Since the Bush represents a bolt, as soon as one of the failure modes occurs (either tension/compression or shear) the entire Bush fails.
7. Unlike PBUSHT implementation in Nastran, the stiffness frequency tables, force per velocity frequency tables, and structural damping frequency tables are not supported.

PCOHE — Interface Cohesive Zone Modeling Element Properties

Defines the properties of a fully nonlinear element used to simulate the onset and progress of delamination.

Format and Example

1	2	3	4	5	6	7	8	9	10
PCOHE	PID	MID		T					
PCOHE	700	701		1.0					

Field	Contents	Type	Default
PID	Property identification.	I > 0	Required
MID	Identification number of a MCOHE entry.	I > 0	Required
T	Initial thickness. See Remark 2.	R > 0	Not used

Remarks:

1. PCOHE can be referenced by CIFHEX.
2. T is required to calculate correct mass when there is no volume in CIFHEX element initially. Element mass will be calculated by

   $$M = \rho A T$$

   where,

   M is element mass,

   ρ is density,

   A is area of collapsed CIFHEX element,

   T is thickness in PCOHE.

   If a CIFHEX element has volume, T will be ignored in element mass calculation.

PCOMP

Layered Composite Element Property

Defines the properties of a multi-ply laminate composite material.

Format and Example

1	2	3	4	5	6	7	8	9	10
PCOMP	PID							LAM	+
PCOMP	181								+
+	MID1	T1	THETA1		MID2	T2	THETA2		+
+	171	0.056	0.				45.		+
+	MID3	T3	THETA3		MID4	T4	THETA4		
+			-45.				90.		

Field	Contents		Type	Default
PID	Unique property number		I ≥ 0	Required
LAM	Symmetric lamination option:		C	Blank
	Blank	Enter all plies.		
	SYM	Describe only plies on one side of the element center line. (See Remark 3.)		
MIDi	Material number of the various plies. Identify the plies by sequentially numbering them starting from 1 at the bottom layer. The MIDs must refer to a MAT1, MAT8, DMATEP, or DYMATzy entry.		I ≥ 0	See Remark 1.
Ti	Thickness of ply i		R ≥ 0	See Remark 1.
THETAi	Orientation angle of the longitudinal direction of each ply with the material axis of the element. (If the material angle on the element connection entry is 0.0, the material axis and side 1-2 of the element coincide.) The plies are numbered sequentially starting with 1 at the bottom layer. (The bottom layer is defined as the surface with the largest negative z-value in the element coordinate system.)		R	0.0

Remarks

1. The default under `MID2`, `MID3`, . . ., is the last-defined material, in this case `MID1`; for `T2`, `T3`, . . ., all these thicknesses are equal to `T1`.
2. At least one of the three values (`MIDi`, `Ti`, `THETAi`) must be present for a ply to exist. The minimum number of plies is one.
3. The symmetric laminate option is currently not available.
4. The thickness of the element is the sum of the ply thicknesses regardless of the values on the CTRIA3 or CQUAD4 Bulk Data entries.

PCOMPA Additional Data for Layered Composite Element Property

Defines additional properties of a multi-ply laminate composite material.

Format and Example

1	2	3	4	5	6	7	8	9	10
PCOMPA	PID	FORM	SHFACT	REF	STRDEF	DT1D	STRNOUT	CLT	
PCOMPA	10	BLT						1	
+	SPINCOR								
+	YES								

Field	Contents		Type	Default
PID	Unique property number referring to a PCOMP property number		I > 0	Required
FORM	Element formulation		C	See Remark 1.
SHFACT	Shear correction factor, see Remark 4.		R	0.83333
REF	Reference surface:		C	MID
	TOP	Reference surface is the top of the surface		
	MID	Reference surface is the central surface		
	BOT	Reference surface is the bottom surface		
STRDEF	Definition in stress-strain output:		C	FIBER
	FIBER	Stresses defined in the fiber and matrix directions.		
	ELEM	Stresses defined in the element coordinate system		
DT1D	Time step skip for one-dimensional failure modes		C	NO
	YES	Skip one-dimensional failure modes		
	NO	Normal time-step calculation		
	See Remark 2.			
STRNOUT	Strain output option		C	YES
	YES	Total strain is calculated		
	NO	No strain is stored in memory		
	See Remarks 3. and 4.			

Field	Contents		Type	Default
ICLT	Option to use Classical Lamination Theory		I	0
	1	Use the Classical Lamination Theory		
	0	Use the integration technique		
	See Remark 4.			
SPINCOR	Spin correction:		C	NO
	NO	No SPINCOR applied		
	YES	SPINCOR applied		
	See Remark 5.			

Remarks

1. For CQUAD4 elements, the default formulation is Key-Hoff. For CTRIA3 elements, the default formulation is C0-TRIA. See the *Dytran User's Guide*, Chapter 5: Application Sensitive Default Settingon application sensitive defaults.
2. If the failure mode is such that fiber and shear strength or matrix and shear strength are lost in all layers, the element is not included in the time-step calculation. If the element fails completely, the element is omitted from the time-step calculations, irrespective of the value entered in this field.
3. If the STRNOUT field is NO, the strain cannot be output.
4. If ICLT is set to 1, the analysis is performed with classical lamination theory (For more detail about the classical lamination theory, see the *Dytran Theory Manual*, Chapter 5: Classical Lamination Theory (CLT) for Multilayered Shells In this case, it is not possible to request the total strain output. The (transverse) shear correction factor input is ignored since it is calculated inside Dytran. There is no update of the cross-sectional properties due to failure. The failure flag only indicates that the failure condition is satisfied. Additional output for element variables is available, namely the stress resultants (NXX, NYY, NXY, MXX, MYY, MXY, QYZ, and QZX). Also the ABD-Q matrices of each element can be requested for output. These data are only stored in the first layer. The variable names are AijM, BijM, DijM, and QsijM for the components of the A-, B-, D- and Q-matrices, respectively. For example, to request the A11 of the A-matrix, the variable name is A11M01.
5. The options for SPINCOR are:

NO	No SPINCOR correction is applied
YES	A SPINCOR correction is applied.

When SPINCOR = NO, slight asymmetric forces are applied to the shell element's grid points. This approach is, in general, acceptable up to about 10° in plane shear angle.

The SPINCOR option is required for fabric models and is turned on by default to accurately keep track of the fiber directions.

PDAMP

Linear Damper Properties

Defines the properties of a linear viscous damper.

Format and Example

1	2	3	4	5	6	7	8	9	10
PDAMP	PID	C							
PDAMP	7	0.01							

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
C	Damping constant (force/velocity or moment/velocity)	R	0.0

Remarks

1. This entry defines a linear viscous damper.
2. For a discussion of the various types of damper elements, see Lagrangian Elements.

PELAS

Elastic Spring Property

Defines the stiffness coefficient, the damping coefficient, and the stress coefficient of an elastic spring.

Format and Example

1	2	3	4	5	6	7	8	9	10
PELAS	PID	K			PID	K			
PELAS	7	4.29			27	2.17			

Field	Contents	Type	Default
PID	Property number	I ≥ 0	Required
K	Spring stiffness	R	0.

Remarks

1. Be cautious when using negative spring-stiffness values because values are defined directly on some of the CELASn entry types.
2. One or two elastic spring properties may be defined on a single entry.
3. For a discussion of the various types of spring elements, see *Dytran User's Guide*, Chapter 2: Elements, Lagrangian Elements.

PELAS1 Nonlinear Elastic Spring with Hysteresis Property

Defines the properties of nonlinear, elastic springs.

Format and Example

1	2	3	4	5	6	7	8	9	10
PELAS1	PID	LOAD	UNLOAD						
PELAS1	5	25	25						

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
LOAD	Number of a TABLExx entry defining the variation of force/moment (y-value) with displacement/ rotation (x-value) during loading.	I > 0	Required
UNLOAD	Number of a TABLExx entry defining the variation of force/moment (y-value) with displacement/rotation (x-value) during unloading.	1 > 0	See Remark 3.

Remarks

1. The values in the table are either force and displacement or moment and rotation, depending on whether the spring connects translational or rotational degrees of freedom.
2. The values in the table are interpolated to determine the force/moment for a particular displacement/rotation.
3. If UNLOAD table is not defined, unloading occurs corresponding to the LOAD curve.
4. Input for loading and unloading must be consistent. Both curves must be either completed defined or have only positive values (start from (0.,0.)). When only positive values are defined, the curves are mirrored automatically.
5. For a discussion of the various types of spring elements, see *Dytran User's Guide*, Chapter 2: Elements, Lagrangian Elements.

PELASEX UDS Properties of an Elastic Spring

Defines the properties of a User Defined Service of an elastic spring.

Format and Example

1	2	3	4	5	6	7	8	9	10
PELASEX	PID	GROUP	PROP1	PROP2	PROP3	PROP4	PROP5	PROP6	
PELASEX	10	USRELAS	1.05						

Field	Contents	Type	Default
PID	Property number	I > 0	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required
PROPi	Property values which can be used in the user subroutine	R	0.0

Remarks

1. The Property ID (PID) must be referenced by a CELAS1 or CELAS2 entry.
2. See the explanation in Chapter 7: User Defined Services on how to use user-written subroutines.
3. The values for PROP(i) will be passed on the User Defined Services in variable PROP(1...6).

PERMCPL — Permeability between Coupling Surfaces

Defines a permeable area of a couple (sub)surface.

The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference.

Format and Example

1	2	3	4	5	6	7	8	9	10
PERMCPL	PID	CSID	PERM-C	PERM-T	FLOW				
PERMCPL	82	10	0.0005		BOTH				

Field	Contents		Type	Default
PID	Unique number of a PERMCPL entry		I > 0	Required
	It is referenced from COUPOR to model the flow between two Euler domains.			
CSID	The ID of the COUPLE entry		I > 0	Required
	This COUPLE is the one that is connected to the coupling surface that references this entry.			
PERM-C	Permeability is a linear function of the pressure difference.		R > 0.0	See Remark 3.
	permeability = PERM – C*abs (Pinside – Pgbid)			
	The gas flow is from the higher to the lower pressure.			
PERM-T	Permeability is a tabular function of the pressure difference:		I > 0	See Remark 3.
	table contains: permeability versus |Pinside – Pgbid|			
	The gas flow is from the higher to the lower pressure.			
FLOW	Defines the allowed directions of the flow.		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed into the COUPLE that references this entry.		
	OUT	Only outflow allowed into the COUPLE that references this entry.		

Remarks

1. The PERMCPL entry can only be referenced from a COUPOR entry.
2. This option can only be used in combination with PARAM, FASTCOUP.
3. The entry can only be used with the single material hydrodynamic Euler solver, using an EOSGAM (ideal gas) equation of state.

4. Either `PERM-C` or `PERM-T` must be specified.

PERMEAB

Air Bag Permeability

Defines the permeability of a couple and/or GBAG (sub)surface.

Permeability is the velocity of gas flow through a (sub)surface and is defined as a linear or tabular function of the pressure difference over the (sub)surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
PERMEAB	PID	PERM-C	PERM-T	FLOW	PENV	RHOENV	SIEENV	CP	
PERMEAB	201	0.5		OUT	1.E5	1.128	2.21E5	1001.	

Field	Contents		Type	Default
PID	Unique number of a PERMEAB entry		I > 0	Required
PERM-C	Permeability is a linear function of the pressure difference.		R > 0.0	See Remark 3.
	permeability = PERM – C*abs (Pinside – PENV)			
	For Pinside > PENV: outflow			
	For Pinside < PENV: inflow			
PERM-T	Permeability is a tabular function of the pressure difference:		I > 0	See Remark 3.
	table contains: permeability versus |Pinside – PENV|			
	For Pinside > PENV: outflow			
	For Pinside < PENV: inflow			
FLOW	Defines the allowed directions of the flow.		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed.		
	OUT	Only outflow allowed.		
PENV	Environmental pressure		R > 0.0	Required
RHOENV	Environmental density		R > 0.0	Required
SIEENV	Environmental specific internal energy		R > 0.0	Required
CP	Environmental specific heat at constant pressure		R > 0.0	See Remark 5.

Remarks

1. The PERMEAB entry can be referenced from a COUPOR and/or GBAGPORentry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler or Roe solver using an EOSGAM (ideal gas) equation of state.
3. Either PERM-C or PERM-T must be specified.

4. The values for the environment $p_{env}(PENV)$, $\rho_{env}(RHOENV)$, $e_{env}(SIEENV)$ must be defined consistent with an ideal gas equation of state:

   $$p_{env} = (\gamma_{env} - 1)\rho_{env} e_{env}$$

   The γ_{env} is calculated by Dytran, and is used when inflow occurs. Inflow occurs when $p_{env} > p_{inside}$.

5. CP is only required if updating of Euler or gas bag gas constants is done, for example if hybrid inflators are defined.

PERMGBG

Air Bag Permeability

Defines a permeable area of a couple and/or GBAG (sub)surface, connected to another GBAG.

The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference.

Format and Example

1	2	3	4	5	6	7	8	9	10
PERMGBG	FID	PERM-C	PERM-T	FLOW	GBID				

Field	Contents		Type	Default
FID	Unique number of a PERMGBG entry		I > 0	Required
	It can be referenced from either a GBAGPOR to model the flow between GBAGs, or from a COUPOR to model the flow between a Eulerian air bag and a COUPOR			
PERM-C	Permeability is a linear function of the pressure difference.		R > 0.0	See Remark 3.
	permeability = PERM-C*abs (Pinside – Pgbid)			
	The gas flow is from the higher to the lower pressure.			
PERM-T	Permeability is a tabular function of the pressure difference.		I > 0	See Remark 3.
	table contains: permeability versus |Pinside – Pgbid|			
	The gas flow is from the higher to the lower pressure.			
FLOW	Defines the allowed directions of the flow.		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed into the GBAG or the coupling surface that references this entry.		
	OUT	Only outflow allowed into the GBAG or the coupling surface that references this entry.		
GBID	Number of a GBAG entry		R > 0.0	Required
	This GBAGis the one that is connected to the GBAG or coupling surface that references this entry.			

Remarks

1. The PERMGBG entry can be referenced from a COUPOR and/or GBAGPOR entry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler solver, using an EOSGAM (ideal gas) equation of state.
3. Either PERM-C or PERM-T must be specified.

PEULER

Eulerian Element Properties

Defines the properties of Eulerian elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
PEULER	PID	MID	TYPE						
PEULER	100	25							

Field	Contents		Type	Default
PID	Unique property number		I > 0	Required
MID	Number of a DMATxxx entry defining the constitutive model		I ≥ 0	Required
TYPE	The type of Eulerian material being used:		C	HYDRO
	HYDRO	Hydrodynamic material with no shear strength + void		
	1stOrder	Single material, 1st order accurate Riemann solution-based fluids and gases Euler solver		
	2ndOrder	Single material, 2nd order accurate Riemann solution-based fluids and gases Euler solver		
	STRENGTH	Structural material with shear strength + void		
	MMHYDRO	Multimaterial hydrodynamic material with no shear strength + void		
	MMSTREN	Structural multimaterial with shear strength + void		

Remarks

1. Make the property number unique with respect to all other property numbers.
2. The elements that reference this property use the Eulerian formulation.
3. If TYPE is set to HYDRO, only one material number for all the Eulerian elements of TYPE is used and a hydrodynamic yield model is chosen.
4. If the TYPE is set to either 1stOrder or 2ndOrder, only one material for all Eulerian elements of TYPE is used and the Riemann solution-based solver is chosen.
5. If TYPE is set to STRENGTH, only one material number for all the Eulerian elements of TYPE is used and a nonhydrodynamic yield model is chosen.
6. If TYPE is set to MMHYDRO, different material numbers for all Eulerian elements of TYPE are used and a hydrodynamic behavior is chosen for each material.
7. If TYPE is set to MMSTREN, different material numbers for all Eulerian elements of TYPE are used and a yield model is chosen for each material.

8. In a multimaterial Euler calculation, the options `MMSTREN` and `MMHYDRO` can not be mixed; they are mutually exclusive.
9. If the material number is blank or zero, the corresponding elements are void. Note that this is not allowed in the Riemann solution-based Euler solvers, as they do not handle void elements. If you define void elements and select either the `1stOrder` or `2ndOrder` scheme, an error message is issued and the analysis stops.
10. Initial conditions are defined on the TICEL Bulk Data entry.

PEULER1 Eulerian Element Properties

Eulerian element properties. The initial conditions of these elements are defined in geometric regions.

Format and Example

1	2	3	4	5	6	7	8	9	10
PEULER1	PID		TYPE	SID	SID2				
PEULER1	100		HYDRO	300	400				

Field	Contents		Type	Default
PID	Unique property number		I > 0	Required
TYPE	The type of Eulerian material(s) being used:		C	HYDRO
	HYDRO	Hydrodynamic material + void		
	1stOrder	Single material, 1st order accurate Riemann solution-based fluids and gases solver		
	2ndOrder	Single material, 2nd order accurate Riemann solution-based fluids and gases solver		
	STRENGTH	Structural material with shear strength + void		
	GTH	Structural material with shear strength + void		
	MMHYDRO	Multimaterial hydrodynamic + void		
	MMSTREN	Structural multimaterial with shear strength + void		
SID	Number of a TICEUL entry specifying the materials and geometric grouping criteria		I > 0	Required
SID2	Number of a BODYFR1 entry defining body force.		I ≥ 0	0

Remarks

1. Remarks 1 through 9 of PEULER apply also here.
2. Initial conditions and/or material assignments are defined on theTICEUL Bulk Data entry.

PLOAD

Pressure Loads on the Face of Structural Elements

Defines a pressure load on a triangular or quadrilateral shell or membrane element or on the face of a Lagrangian solid element.

Format and Example

1	2	3	4	5	6	7	8	9	10
PLOAD	LID	SCALE	G1	G2	G3	G4			
PLOAD	1	-4.0	16	32	11				

Field	Contents	Type	Default
LID	Load set number	I > 0	Required
SCALE	Scale factor for the pressure	R	1.0
G1-G4	Grid-point numbers defining either a triangular or quadrilateral surface to which the pressure is applied. For a triangular surface, G4 is blank or zero.	I > 0	Required

Remarks

1. For quadrilateral surfaces, order the grid points G1 through G4 around the perimeter of the surface, and number them clockwise or counterclockwise.
2. The direction of positive pressure is calculated according to the right-hand rule using the sequence of grid points. See *Dytran User's Guide*, Chapter 2: Elements, Lagrangian Elements.
3. Reference LID from a TLOADn entry.
4. The pressure $p(t)$ at time t is given by

   $$p(t) = SCALE*T(t)$$

   where $SCALE$ is the scale factor and $T(t)$ is the value interpolated from the function or table given on the TLOADn entry at time t.

PLOAD4 Pressure Loads on the Face of Structural Elements

Defines a load on a face of a CHEXA, CPENTA, CTETRA, CTRIA3, or CQUAD4 element.

Format and Example

1	2	3	4	5	6	7	8	9	10
PLOAD4	LID	EID	P1				G1	G3/G4	
PLOAD4	2	1106	10.0				48	123	

Field	Contents	Type	Default
LID	Load set number.	I > 0	Required
EID	Element number.	I > 0	Required
P1	Load per unit surface (pressure) on the face of the element.	R	Required
G1	Number of a grid point connected to a corner of the face. Required data for solid element only (integer or blank).	I > 0	Required
G3	Number of a grid point connected to a corner diagonally opposite to G1 on the same face of a CHEXA or CPENTAelement. Required data for quadrilateral faces of CHEXAand CPENTA elements only (integer or blank). G3 is omitted for a triangular surface on a CPENTA element.	I > 0	Required
G4	Number of the CTETRA grid point located at the corner not on the face being loaded. This is required data and is used for CTETRA elements only.	I > 0	Required

Remarks

1. For solid (CTETRA, CPENTACTETRA) elements, the direction of positive pressure is inwards.
2. For plate elements, (CQUAD4, CTRIA3 the direction of positive pressure is the direction of the positive normal determined by applying the right-hand rule to the sequence of the element grid-point connectivity.
3. G1 and G3 are ignored for CTRIA3 and CQUAD4 elements.
4. For the triangular faces of CPENTA elements, G1 is a corner grid-point number that is on the face being loaded, and the G3 or G4 field is left blank. For the faces of CTETRA elements, G1 is a corner grid-point number that is on the face being loaded, and G4 is a corner grid-point number that is not on the face being loaded. Since a CTETRA has only four corner grid points, this grid point G4 is unique and different for each of the four faces of a CTETRA element.
5. If the pressure is 9999., a pressure load is not applied. Instead, it is translated to a CFACE1 entry. This makes it easy to generate CFACE1 entries using a standard preprocessor. See *Dytran User's Guide*, Chapter 9: Running the Analysis,Using a Modeling Program with Dytran for details. The LID field is converted to the number of the set of faces.
6. Reference LID by a TLOAD Bulk Data entry.

7. The pressure $p(t)$ at time t is given by:

   $$p(t) = SCALE*T(t)$$

   where $SCALE$ is the scale factor and $T(t)$ is the value interpolated from the function or table given on the TLOADn entry at time t.

PMARKER

Property Definition of a Marker Element

Defines the behavior of the marker elements in the FV domain.

Format and Example

1	2	3	4	5	6	7	8	9	10
PMARKER	ID	TYPE							
PMARKER	7	FIXED							

Field	Contents		Type	Default
ID	Marker property ID; referred to by CMARK# entries		I > 0	Required
TYPE	Behaviour of the marker grid points in the FV domain		C	FIXED
	FIXED	the marker does not move in the FV domain.		
	MOVING	the marker is moved by velocities in the FV domain.		

Remarks

1. The PMARKER entry is ignored for elements referring to structural grid points. These structural grid points move with the structure and the FV velocities do not change their velocity.
2. Type = FIXED: means that the marker is stationary through out the simulation and is, therefore, not moving with the Euler velocity. If the marker grid is located outside the Eulerian domain(s), the Marker is allowed to exist. However, no variables are recorded and they appear as zero on the Time History plots.
3. Type = Moving: marker is moving along with the Eulerian material. When the grid point approaches a coupling surface, there is no mechanism that prevents the marker from passing through the coupling surface. When this happens, the marker enters an element that is covered and motion of the grid point stops. It is allowed that the grid point moves from one Euler domain to the other through a porous hole or a coupling surface with interactive failure.

PMINC Constant Spallation Model

Defines a spallation model where the minimum pressure is constant.

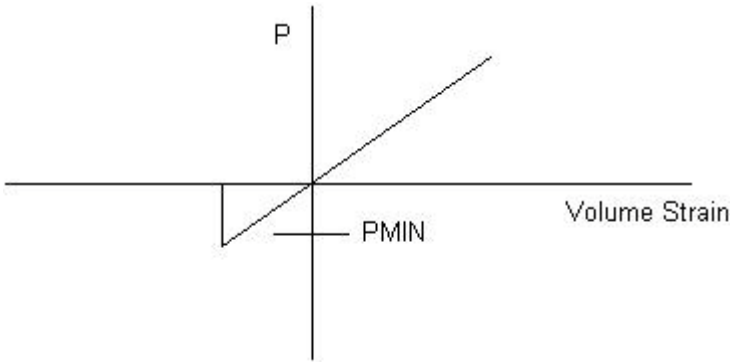
Format and Example

1	2	3	4	5	6	7	8	9	10
PMINC	PID	VALUE	FVTOL	FVTOL2					
PMINC	220	-370.							

Field	Contents	Type	Default
PID	Unique PMINC number	I > 0	Required
VALUE	The value of the minimum pressure	R ≤ 0.0	See Remark 2.
FVTOL	Void fraction cutoff tolerance	R > 0.0	1.E-4.
FVTOL2	Maximal void fraction that is permissible under tension.	R>0.0	0 See Remark 4.

Remarks

1. If the pressure in an element falls below the minimum pressure, the element spalls. The pressure and yield stress are set to zero.



2. The default for the minimum pressure for Lagrangian solids is -1.E20. For Eulerian elements, the default is 0.0.
3. If an element spalls a void is created. To prevent too small void fractions, a void fraction is put to zero if it is smaller than FVTOL. The default for FVTOL is 1.E-4 and works only for Eulerian elements. This value should be decreased in case of large mass increase of material without any reason. Voids can be created during transport of material, because of a material failure and by unloading.

4. With FVTOL2 =0, any void fraction in an element will lead to failure, and then no tensile stresses are possible. In simulations in which tensile conditions are present, it can be required to allow for tensile stresses in the presence of a small void fraction not exceeding a threshold. This threshold is given by FVTOL2. A good value for FVTOL2 = 2*FVTOL = 2.E-4. FVTOL2 is only used for Eulerian materials. FVTOL2 should be larger than FVTOL.

POREX User-defined Porosity Model Specified by a User Subroutine

Defines a porosity model through a user-written subroutine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE mypor SCA.MDSolver.Obj.Uds.Dytran.Flow
```

Format and Example

1	2	3	4	5	6	7	8	9	10
POREX	PID	NAME	GROUP						
POREX	7	MYFLOW	mypor						

Field	Contents	Type	Default
FID	Unique POREX number	I > 0	Required
NAME	Name of the porosity model. See Remark 3.	C	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. The porosity ID (PID) must be referenced by a COUPOR entry.
2. See the explanation in Chapter 7: User Defined Services on how to use user-written subroutines.
3. The PID of the POREX entry is passed to the user-written subroutine and can be used to identify the porosity model.

PORFCPL Flow Between Two Coupling Surfaces Through a Hole

Defines an interaction between two coupling surfaces through a small hole. The velocity of the gas flow through the hole is based on the pressure method.

Format and Example

1	2	3	4	5	6	7	8	9	10
PORFCPL	PID			FLOW	CSID				
PORFCPL	1			BOTH	1				

Field	Contents		Type	Default
PID	Unique PORFCPL ID		I > 0	Required
FLOW	Defines the allowed directions of the flow:		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed into the COUPLE that references this entry.		
	OUT	Only outflow allowed into the COUPLE that references this entry.		
CSID	The ID of the COUPLEentry		I > 0	Required
	This COUPLE is the one that is connected to the coupling surface that references this entry			

Remarks

1. The PORFCPL entry can only be referenced from COUPOR entry.
2. This option can only be used in combination with PARAM, FASTCOUP, and EOSGAM material.
3. The pressure method used to calculate the material transport through a porous (sub) surface is described in the *Dytran User's Guide*, Chapter 4: Fluid Structure Interaction, General CouplingFor more detail on modeling flow between Eulerian domains, see PARAM, FLOW-METHOD.

PORFGBG

Flow Between Two Air Bags Through a Hole

Defines a hole in a couple and/or GBAG (sub)surface, connected to another GBAG.

The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice, and depends on the pressure difference.

Format and Example

1	2	3	4	5	6	7	8	9	10
PORFGBG	FID			FLOW	GBID				

Field	Contents		Type	Default
FID	Unique number of a PORFGBG entry.		I > 0	Required
	It can be referenced from either a GBAGPOR to model the flow between GBAGs, or from a COUPOR to model the flow between an Eulerian air bag and a GBAG.			
FLOW	Defines the allowed directions of the flow:		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed into the GBAG or the coupling surface that references this entry.		
	OUT	Only outflow allowed into the GBAG or the coupling surface that references this entry.		
GBID	Number of a GBAG entry.		I > 0	Required
	This GBAG is the one that is connected to the GBAG or coupling surface that references this entry.			

Remarks

1. The PORFGBG entry can be referenced from a COUPOR and/or GBAGPORentry,
2. When used with Euler, this entry can only be used with the single material hydrodynamic Euler solver, using an EOSGAM (ideal gas) equation of state.

PORFLCPL Flow Between Two Coupling Surfaces Through a Large Hole

Defines an interaction between two coupling surfaces through a large hole. The velocity of the gas flow through the hole is based on the velocity method.

Format and Example

1	2	3	4	5	6	7	8	9	10
PORFLCPL	PID			FLOW	CSID				
PORFLCPL	1			BOTH	1				

Field	Contents		Type	Default
PID	Unique PORFLCPL ID		I > 0	Required
FLOW	Defines the allowed directions of the flow:		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed into the COUPOR that references this entry.		
	OUT	Only outflow allowed into the COUPLE that references this entry.		
CSID	The ID of the COUPLE entry.		I > 0	Required
	This COUPLE is the one that is connected to the coupling surface that references this entry.			

Remarks

1. The PORFLCPL entry can only be referenced from COUPOR or GBAGPOR entry.
2. This option can only be used in combination with PARAM, FASTCOUP.
3. The velocity method used to calculate the material transport through a porous (sub) surface is described in *Dytran User's Guide*, Chapter 4: Fluid Structure Interaction, General Coupling.
4. In airbag simulations, very small densities can cause instabilities. Therefore, transport of mass from an Euler element with mass of one coupling surface into an empty Euler element of another coupling surface has to be avoided. Since airbag simulations are mostly run with the single material hydro solver, this solver blocks flow into an empty element through a segment with **PORFLCPL**-porosity. If this flow into empty elements is required, then the multi-material Euler solver has to be used. This solver does not block flow and allows mass to flow into an empty element through a segment with **PORFLCPL**-porosity.

PORFLGBG Flow Between Two Air Bags Through a Large Hole

Defines a hole in a couple and/or GBAG (sub)surface, connected to another GBAG.

The velocity of the gasflow through the hole is based on the velocity method for an Eulerian air bag.

Example and Format

1	2	3	4	5	6	7	8	9	10
PORFLGBG	FID			FLOW	GBID	MID			
PORFLGBG	81			BOTH	20	4			

Field	Contents		Type	Default
FID	Unique number of a PORFLGBG entry.		I > 0	Required
	It can be referenced from either a GBAGPORto model the flow between GBAGs, or from a COUPOR to model the flow between an Eulerian air bag and a GBAG.			
FLOW	Defines the allowed directions of the flow:		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed into the GBAG or the coupling surface that references this entry.		
	OUT	Only outflow allowed into the GBAGor the coupling surface that references this entry.		
GBID	Number of a GBAG entry		I > 0	Required
	This GBAG is the one that is connected to the GBAG or coupling surface that references this entry.			
MID	Material number of the GBAG gas. Used only when connecting a GBAG to a Eulerian air bag that uses the Multi-material Euler solver.		1. 0	See Remark 2.

Remarks

1. The PORFLGBG entry can be referenced from a COUPOR and/or GBAGPORentry,
2. Once gas from a GBAG enters a Eulerian domain, it is treated as Eulerian material. For the single material Euler solver, only one Eulerian material is present and the material number, MID, can be left blank. Since GBAG material is an ideal gas, it is required that Eulerian material also uses an EOSGAM (ideal gas) equation of sate. When using the multi-material solver, the material number, MID, has to point to one of the Eulerian materials and the equation of state of that material has to be of type EOSGAM.
3. The velocity method used to calculate the material transport through a porous (sub)surface are described in *Dytran User's Guide*, Chapter 4: Fluid Structure Interaction, General Coupling.

4. For flow between two uniform pressure air bags, the material transport is based on the theory of one-dimensional gas flow through a small orifice, and depends on the pressure difference. This is equivalent to the PORFLGBG entry.

PORFLOW

Porous Flow Boundary

Defines the material properties for the in- or outflow of an Eulerian mesh through a porous (SUBSURF)SURFACE.

Format and Example

1	2	3	4	5	6	7	8	9	10
PORFLOW	FID		TYPE1	VALUE1	TYPE2	VALUE2	TYPE3	VALUE3	+
PORFLOW	120		XVEL	100.0					+
+	TYPE4	VALUE4	-etc.-						
+									

Field	Contents		Type	Default
FID	Unique number of a PORFLOW entry		I > 0	Required
TYPEi	The properties on the flow boundary:		C	Required
	MATERIAL	Material number		
	XVEL	Velocity in the x-direction		
	YVEL	Velocity in the y-direction		
	ZVEL	Velocity in the z-direction		
	PRESSURE	Pressure		
	DENSITY	Density		
	SIE	Specific internal energy		
	FLOW	The type of flow boundary required		
	METHOD	The method used for the material transport		
VALUEi	The value of the property specified in the TYPE field		R or C	Required
	For TYPEi set to FLOW, the value is a character entry: either IN, OUT, or BOTH, indicating that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH.			See Remark 4.
	For TYPEi set to METHOD, the value is a character entry: either VELOCITY or PRESSURE, indicating that the material transport is based on the velocity method or the pressure method. The default is VELOCITY.			See Remark 4.

Remarks

1. Reference FID by a COUPOR entry.
2. Any material properties not specifically defined have the same value as the element that the c segment is intersecting.
3. The SURFACE can be only a general coupling surface (see the COUPLE entry).
4. The different methods used to calculate the material transport through a porous (sub)surface are described in *Dytran User's Guide*, Chapter 4: Fluid Structure Interaction, General Coupling.
5. METHOD=VELOCITY is valid for all equation of state models.

   METHOD=PRESSURE is valid for EOSGAM (ideal gas) in combination with the single material hydrodynamic Euler solver.
6. Alternative methods are available to define holes and permeable sections in an air bag. See the entries: COUPOR, GBAGPOR, PORHOLE, PERMEAB, PORFGBG and PERMGBG
7. In the case of material flow into a multi-material Euler mesh, the material number, the density, and specific energy have to be set. On the other hand, when material flows out of a multi-material Euler mesh, it is assumed that each of the materials present in the outflow Euler element contributes to the outflow of mass. The materials are transported in porportion to their relative volume fractions.
8. Prescribing both pressure and velocity may lead to the instabilities.

PORFLOWT

Time-dependent Porous Flow Boundary

Definition of a time dependent flow through a porous (SUBSURF)SURFACE

Format and Example

1	2	3	4	5	6	7	8	9	10
PORFLOWT	FID	TYPE							+
PORFLOWT	2	IN							+
+	VELTYPE	VELOCITY	PRESTYP	PRES					
+	TABLE	101	TABLE	102					
+	MID	DENSTYP	DENSITY	SIETYPE	SIE				
+	91	TABLE	104	TABLE	107				

Field	Contents		Type	Default
FID	Unique number of a PORFLOW entry		I > 0	Required
TYPE	IN	Inflow boundary (see Remarks 2. and 3.)	C	Both
		Only inflow is allowed. The inflow velocity and pressure can be optionally specified. If not given, the values in the adjacent Euler element is used. The same holds for the DENSITY and SIE.		
	OUT	Only outflow is allowed. The inflow velocity and pressure can be optionally specified. If not given, the values in the adjacent Euler element are used. The outflow boundary always uses the material mixture present in the adjacent Euler element.		
	BOTH	Material is allowed to flow in or out. In- or outflow is based on the direction of the velocity in the adjacent Euler element. Only pressure can be optionally defined. If not given, the pressure in the adjacent Euler element is taken.		
VELTYPE	Type of velocity definition:		C	Element
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		

Field	Contents		Type	Default
	TABLE	Value varies in time		
VELOCITY	Value of inflow or outflow velocity. If VELTYPE = TABLE, it refers to a TABLED1 orTABLEEX ID. The velocity direction is normal to the coupling surface or subsurface. A positive velocity corresponds with inflow.		I or R	See Remark 7.
PRESTYP	Type of pressure definition:		C	See Remark 7.
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		
	TABLE	Value varies in time		
PRES	Value of inflow or outflow pressure. If PRESTYPE = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	
MID	Material ID of inflowing material. Input is not allowed for TYPE = OUT.		I	
	When MID is specified, it is required to also define density and SIE for the inflowing material.			
DENSTYP	Type of density definition:		C	Required when MID is given.
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		
	TABLE	Value varies in time		
DENSITY	Value of density. If DENSTYP = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	Required when MID is given.
SIETYPE	Type of density definition		C	Required when MID is given.
	ELEMENT	Value of Euler element		
	CONSTANT	Value is constant in time		
	TABLE	Value varies in time		
SIE	Value of density. If SIETYPE = TABLE, it refers to a TABLED1 or TABLEEX ID.		I or R	Required when MID is given.

Remarks

1. Reference FID by a COUPOR entry.

2. Any material properties not specifically defined have the same value as the element that the (SUBSURF)SURFACEsegment is intersecting.
3. The SURFACE can be only a general coupling surface (see the COUPLE entry).
4. The different methods used to calculate the material transport through a porous (sub)surface are described in *Dytran User's Guide*, Chapter 4: Fluid Structure Interaction, General Coupling. These are METHOD=VELOCITY and METHOD=PRESSURE. For PORFLOWT, the VELOCITY method is used. The PRESSURE method is not available.
5. Alternative methods are available to define holes and permeable sections in an air bag. See the entries: COUPOR, GBAGPOR, PORHOLE, PERMEAB, PORFGBG, and PERMGBG.
6. In the case of material flow into a multi-material Euler mesh, the material number, the density, and specific energy have to be set. On the other hand, when material flows out of a multi-material Euler mesh, it is assumed that each of the materials present in the outflow Euler element contributes to the outflow of mass. The materials are transported in porportion to their relative volume fractions.
7. The boundary condition initiates or determines a wave in compressible material like gas and water. This can be either an outgoing or an ingoing wave. For stability, it is important that the waves created are compatible with the flow type near the boundary. Relevant flow types are subsonic inflow, subsonic outflow, supersonic inflow, and supersonic outflow. For example, for subsonic inflow, prescribing both pressure and velocity would initiate outgoing waves. Outgoing waves for an inflow boundary condition is known to be instable. However, for supersonic inflow, you can specify both pressure and velocity since there are no outgoing waves at a supersonic inflow boundary.
8. When a TABLEEX is referenced, the EXFUNC user subroutine must be created. See TABLEEX for more details.

PORHOLE

Holes in Air Bag Surface

Defines a hole in a couple and/or GBAG (sub)surface.

The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice and depends on the pressure difference.

Format and Example

1	2	3	4	5	6	7	8	9	10
PORHOLE	FID			FLOW	PENV	RHOENV	SIEENV	CP	
PORHOLE	301				0.1	1.1E-12	2.2E11		

Field	Contents		Type	Default
PID	Unique number of a PORHOLE entry		I > 0	Required
FLOW	Defines the allowed directions of the flow:		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed.		
	OUT	Only outflow allowed.		
PENV	Environmental pressure		R > 0.0	Required
RHOENV	Environmental density		R > 0.0	Required
SIEENV	Environmental specific internal energy		R > 0.0	Required
CP	Environmental specific heat at constant pressure		R > 0.0	See Remark 4.

Remarks

1. The PORHOLE entry can be referenced from a COUPOR and/or GBAGPORentry.
2. When used with Euler, this entry can only be used with the single material hydrodynamic Euler solver, using an EOSGAM (ideal gas) equation of state.
3. The values for the environment p_{env} (PENV), ρ_{env} (RHOENV), e_{env} (SIEENV) must be defined consistent with an ideal-gas equation of state:

   $$p_{env} = (\gamma_{env} - 1)\rho_{env} e_{env}$$

   The γ_{env} is calculated by Dytran, and is used when inflow occurs. Inflow occurs when $p_{env} > p_{inside}$.
4. CP is only required if updating of Euler or gas bag gas constants is done, for example if hybrid inflators are defined.

PORHYDST

Porous Flow Boundary

Prescribes a hydrostatic pressure profile on a porous (SUB)SURFACE.

Format and Example

1	2	3	4	5	6	7	8	9	10
PORHYDST	FID								
PORHYDST	120								

Field	Contents	Type	Default
FID	Unique number of a PORHYDST entry	I > 0	Required

Remarks

1. Reference FID by a COUPOR entry.
2. The velocity and outflow density have the same value as the element that the (SUB)SURFACEsegment is intersecting.
3. The SURFACE can be only a general coupling surface (see the COUPLE entry).
4. It is required that the coupling surface refers to a HYDSTAT entry. This HYDSTAT entry will be used to prescribe a hydrostatic pressure profile on the subsurface. For example, the water level and atmospheric pressure are taken from the HYDSTAT entry. This defines the pressure and the inflow density.
5. In contributions of the surface to the Euler elements, the pressure gradient across the surface is taken into account. Therefore, splitting up of the surface and creating new PORHYDST entries does not increase the accuracy of the prescribed pressures. If the water level and atmospheric pressure are the same in the whole region outside the coupling surface, using one PHORHYDST entry is sufficient.
6. The atmospheric pressure is prescribed on those parts of the surface that are above the water level.

PORLHOLE

Large Hole in Air Bag Surface

Defines a hole in a couple and/or GBAG (sub)surface.

The velocity of the gasflow through the hole is based on the velocity method for an Eulerian air bag.

Format and Example

1	2	3	4	5	6	7	8	9	10
PORLHOLE	FID			FLOW	PENV	RHOENV	SIEENV	CP	+
PORLHOLE	301				0.1	1.1E-12	2.2E11		+
+	MID								
+	3								

Field	Contents		Type	Default
PID	Unique number of a PORLHOLE entry		I > 0	Required.
FLOW	Defines the allowed directions of the flow:		C	BOTH.
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed.		
	OUT	Only outflow allowed.		
PENV	Environmental pressure		R > 0.0	Required.
RHOENV	Environmental density		R > 0.0	Required.
SIEENV	Environmental specific internal energy		R > 0.0	Required.
CP	Environmental specific heat at constant pressure		R > 0.0	See Remark 4.
MID	Material number of the environmental material		1 > 0	See Remark 2.

Remarks

1. The PORLHOLE entry can be referenced from a COUPOR and/or GBAGPOR entry.
2. When used in combination with the single material hydrodynamic Euler solver, an EOSGAM (ideal gas) equation of state is required. In that case, the material number, MID, can be left blank. When using the multi-material solver, the material number, MID, has to point to one of the Eulerian materials and the equation of state of that material has to be of type EOSGAM.
3. The values for the environment p_{env} (PENV), ρ_{env} (RHOENV), e_{env} (SIEENV) must be defined consistent with an ideal-gas equation of state:

$$p_{env} = (\gamma_{env} - 1)\rho_{env} e_{env}$$

The γ_{env} is calculated by Dytran, and is used when inflow occurs. Inflow occurs when $p_{env} > p_{inside}$.

4. CP is only required if updating of Euler or gas bag gas constants is done, for example if hybrid inflators are defined.
5. The velocity method used to calculate the material transport through a porous (sub)surface are described in *Dytran User's Guide*, Chapter 4: Fluid Structure Interaction, General Coupling.
6. For in and out flow of an uniform pressure air bag (GBAG), the material transport is based on the theory of one-dimensional gas flow through a small orifice, and depends on the pressure difference. This is equivalent to the PORHOLE entry.

PROD

Rod Property

Defines the properties of a rod that is referenced by the CROD entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
PROD	PID	MID	A						
PROD	17	23	42.6						

Field	Contents	Type	Default
PID	Property number	I > 0	Required
MID	Material number	I > 0	Required
A	Cross-sectional area of the rod	R > 0.0.	Required

Remark

All PROD entries must have unique property numbers.

PSHELL

Shell-Element Properties

Defines the properties of shell elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
PSHELL	PID	MID1	T	MID2		MID3			+
PSHELL	10	100	0.01	101		102			+
+			MID4					ICLT	
+			103						

Field	Contents	Type	Default
PID	Unique property number referring to a PSHELL property number	I > 0	Required
MID1	Material number, see Remark 4. and 7.	I ≥ 0	PID
T	Default value for element thickness	R ≥ 0	See Remark 5.
MID2	Material number for bending	I ≥ 0	See Remark 7.
MID3	Material number for transverse shear	I ≥ 0	See Remark 7.
MID4	Material number for membrane-bending coupling	I ≥ 0	See Remark 7.
ICLT	Option to use classical lamination theory, see Remark 7.	I	0

Remarks

1. The property number must be unique with respect to all other properties.
2. Shell of constant thickness with three-point Gauss integration and a transverse shear correction factor of 0.83333 are assumed. For shells using the classical lamination theory option no shear correction factor is applied. The transverse shear stiffness is input as material property using a MAT2 entry. For CQUAD4 elements, the formulation is Key-Hoff and for CTRIA3 elements the formulation is C0-TRIA.
3. If the thickness is set to 9999, all the elements with this property ID are not treated as CQUAD4and CTRIA3 elements but are converted to CSEGentries. This allows CSEG to be defined easily using standard preprocessors. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Modeling of Surfaces and Faces for details.
4. Material entries that can be referenced by shell elements defined on the PSHELL entry can be found in *Dytran Theory Manual*, Chapter 3: Materials.
5. If the thickness is set to blank or 0.0, the thickness on the CTRIA3 and CQUAD4 must be defined.
6. See also *Dytran User's Guide*, Chapter 5: Application Sensitive Default Setting.

7. If ICLT is set to 1, the shells will be analyzed using the classical lamination theory. In this case MID1, MID2, MID3 and MID4 refer to membrane, bending, transverse shear and membrane-bending coupling materials, respectively. All these materials must be filled in and they refer to a MAT2 entry. In this case, the material angles in the element connectivity entries are ignored. More information about Classical Lamination Theory for Shells can be found in the *Dytran Theory Manual,* Chapter 5: Classical Lamination Theory (CLT) for Multilayered Shells

   The element outputs are NXX, NYY, NXY, MXX, MYY, MXY, QYZ, and QZX. These are general forces per unit length. For more detail description about these forces, see the *Dytran Theory Manual*.

PSHELL1

Shell-Element Properties

Defines the properties of Lagrangian shell elements that are much more complicated than the shell elements defined using the PSHELL entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
PSHELL1	PID	MID	FORM	QUAD	NUMB	SHFACT	REF	SPINCOR	+
PSHELL1	7	2	BLT	GAUSS	5	0.9	MID	YES	+
+	T1	T2	T3	T4	TRANSHR	SHRLCK	ADDRES	LENVEC	
+	10.0	10.0	10.0	10.0					

Field	Contents		Type	Default
PID	Unique property number		I > 0	Required
MID	Material number. See Remark 2.		I ≥ 0	PID
FORM	Shell formulation:		C	See Remark 3.
	HUGHES	Hughes-Liu		
	BLT	Belytschko-Lin-Tsay		
	KEYHOFF	Key-Hoff		
	C0-TRIA	C0 triangle		
	MEMB	Membrane element (no bending)		
	DUMMY	Dummy element		
QUAD	Type of quadrature:		C	GAUSS
	GAUSS	Gauss quadrature		
	LOBATTO	Lobatto quadrature		
NUMB	The number of integration points through the thickness. For Gauss and Lobatto quadrature:		I > 0	3
	1	1 point (membrane element)		
	2	2 point		
	3	3 point		
	4	4 point		
	5	5 point		
SHFACT	Shear factor		R	0.83333

Field	Contents		Type	Default
REF	Reference surface (see Remark 14.):		C	MID
	TOP	Reference surface is the top surface.		
	MID	Reference surface is the central surface.		
	BOT	Reference surface is the bottom surface.		
SPINCOR	Spin correction		C	See Remark 13.
	NO	No SPINCOR applied		
	YES	SPINCOR is applied		
T1...T4	Element thickness at the grid points		R ≥ 0.0	See Remark 8.
TRANSHR	Method of transverse-shear calculation:		C	See Remark 10.
	LINEAR	Linear transverse shear		
	CONSTANT	Constant transverse shear		
	CONAPX	Approximated constant transverse shear		
SHRLCK	Shear-lock avoidance:		C	See Remark 10.
	AVOID	Avoid shear lockup		
	NOAVOID	No avoid		
ADDRES	Stores grid-point addresses in memory		C	See Remark 10.
	SAV	Save addresses.		
	NOSAVE	Do not save.		
LENVEC	Vector length		I	See Remark 10.

Remarks

1. Shells of constant thickness with three-point Gauss integration are more easily defined using the PSHELL entry.
2. Material entries that can be referenced by shell elements can be found in Materials.
3. For CQUAD4 elements, the default formulation is KEYHOFF. For CTRIA3 elements, the default formulation is CO-TRIA. Formulations HUGHES, KEYHOFF, and BLT are not allowed for CTRIA3 elements. See also *Dytran User's Guide*, Chapter 5: Application Sensitive Default Settings.
4. Make the property number unique with respect to all other properties.
5. If the thickness T is set to 9999., all elements with this property number are not treated as CQUAD4 and CTRIA3 elements but are converted to CSEG entries. This conversion allows CSEGs to be defined easily using standard preprocessors. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Using a Modeling Program with Dytran for details.
6. Membrane elements can only be triangular and must reference a DMAT or DMATEL material entry. In case the HUGHES shell formulation is used, only an elastic material can be referred to.

7. Dummy elements are used to define rigid bodies or to achieve a closed volume when defining coupling surfaces. Do not use them to create CSEG entries.
8. If the thickness is set to blank or 0.0, the thickness is defined on the CTRIA3and CQUAD4 entry.
9. Specifying QUAD and NUMB is not necessary if FORM is MEMB.
10. The following defaults apply:

	BLT	HUGHES	KEYHOFF
TRANSHR	Not Available	Not Available	LINEAR
SHRLCK	NOAVOID	Not Available	AVOID
ADDRES	SAVE	Not Available	SAVE
LENVEC	LENVEC	Not Available	LENVEC

11. When shell elements undergo large twisting, the linear transverse shear calculations must be used (TRANSHR). It increases accuracy at the expense of more computer time.
12. The default vector length for vector machines is used whenever LENVEC is not defined. Increasing the vector length is usually more efficient, but requires more memory. In some problems a recurrence in the force update may inhibit vectorization on vector machines. Decreasing the vector length may avoid the recurrence. Examine the problem output for information on this recurrence.
13. The options for SPINCOR are:

    NO No SPINCOR correction is applied

    YES A SPINCOR correction is applied.

    When SPINCOR = NO, slight asymmetric forces are applied to the shell element's grid points. This approach is, in general, acceptable up to about 10° in plane shear angle.

    The SPINCOR option is required for fabric models and is turned on by default to accurately keep track of the fiber directions.
14. REF defines the location of the integration pints in the thickness direction. An example for three integration points (x):

```
                          x
REF=MID G1 *----------x-----------* G2
                          x

REF=TOP G1 *----------x-----------* G2
                          x
                          x

                          x
                          x
REF=BOT G1 *----------x-----------* G2
```

PSOLID

Lagrangian Solid-Element Properties

Defines the properties of Lagrangian solid elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
PSOLID	PID	MID		IN		ISOP			
PSOLID	2	100							

Field	Contents		Type	Default
PID	Unique property number		I > 0	Required
MID	Material number. See Remark 2.		I > 0	PID
IN	Integration network		1 > = 0,C	See Remark 3.
	1 or ONE	use one integration network.		
	2 or TWO	use two integration network.		
ISOP	Integration scheme		1 > = 0,C	See Remark 3.
	0 or REDUCED	use reduced integration scheme.		
	1 or FULL	use full integration scheme.		

Remarks

1. The property number must be unique with respect to all other property numbers.
2. Material entries that can be referenced by Lagrangian solid elements are given in *Dytran Theory Manual*,Chapter 3: Materials.
3. IN is a sort of minimum number of integration points in which the element does not have zero energy displacement patterns. For CHEXA (CPENTA, it is 1 x 1 x 1. Therefore, the right value for IN is 1 or 2. The ISOP option is for choosing how the element strains should eventually be calculated. For now, when IN = 2, the right option for ISOP is 1 for full integration and 0 for reduced integration. When IN = 1, the correct option for ISOP is 1.

   The default for CTETRA uses linear tetrahedron FE scheme (IN = 1 and ISOP = 1). The collapsed hexahedron scheme for CTETRA (IN = 2 and ISOP = 0) is deactivated. Please use PARAM , OLDLAGTET , 1 to activate it. If PARAM is set, then old scheme for CTETRA is default. But, it is still possible to use the new TET by using a separate PSOLID with the right IN =1 and ISOP = 1 combination.

The default for CHEXA and CPENTA uses reduced integration scheme (`IN = 2` and `ISOP = 0`). `IN = 2` or `TWO` and `ISOP = 1` or `FULL` means using two integration network. If other combinations are used, they are set to default.

Only limited features are supported for elements with the `IN = 2` and `ISOP = FULL` scheme. The current version supports only MATRIG,DMATEL, DMAT-EOSPOL-SHREL-`YDLVM`, and DMAT-EOSPOL-SHRLVE.

To activate the CTETRA element based on linear tetrahedrom FE formulation, use `IN = 1` and `ISOP = 1`.

4. Use the PEULER entry to define the properties of the Eulerian elements.

PSPR

Linear-Elastic Spring Properties

Defines the properties for a linear-elastic spring with failure.

Format and Example

1	2	3	4	5	6	7	8	9	10
PSPR	PID	K	FAILMTF	FAILMCF					
PSPR	8	20.0E3							

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
K	Elastic stiffness (force/displacement)	R > 0.0	Required
FAILMTF	Tensile failure force	R > 0.0	No failure
FAILMCF	Compressive failure force	R > 0.0	No failure

Remarks

1. This entry defines a linear-elastic spring with failure. PSPR1 can be used to define nonlinear springs.
2. The behavior of this spring is discussed in *Dytran User's Guide*, Chapter 2: Elements, Lagrangian Elements.

PSPR1 Nonlinear-Elastic Spring with Hysteresis Properties

Defines the properties for a nonlinear spring where the stiffness varies with displacement.

Format and Example

1	2	3	4	5	6	7	8	9	10
PSPR1	PID	LOAD	UNLOAD						
PSPR1	8	15	15						

Field	Contents	Type	Default
PID	Unique property number	I > 0	Required
LOAD	Number of a TABLExx entry defining the variation of force (y-value) with displacement (x-value) during loading.	I > 0	Required
UNLOAD	Number of a TABLExx entry defining the variation of force (y-value) with displacement (x-value) during unloading.	1 > 0	See Remark 2.

Remarks

1. The values in the table are interpolated to determine the force for a particular displacement.
2. If the UNLOAD table is not defined, unloading occurs corresponding to the LOAD curve.
3. Input for loading and unloading must be consistent. Both curves must be either completed defined or have only positive values (start from (0.,0.)). When only positive values are defined, the curves are automatically mirrored.
4. The behavior of this spring is discussed in *Dytran User's Guide*, Lagrangian Elements.

PSPREX UDS Properties of a Linear Elastic Spring

Defines the properties of a User Defined Service of a linear elastic spring.

Format and Example

1	2	3	4	5	6	7	8	9	10
PSPREX	PID	GROUP	PROP1	PROP2	PROP3	PROP4	PROP5	PROP6	
PSPREX	10	USRSPR	50.0E3	12.4					

Field	Contents	Type	Default
PID	Property number	I > 0	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required
PROPi	Property values which can be used in the user subroutine	R	0.0

Remarks

1. The Property ID (PID) must be referenced by a CSPR entry.
2. See the explanation in Chapter 7: User Defined Services on how to use user-written subroutines.
3. The values for PROP(i) will be passed on the User Defined Services in variable PROP(1...6).

PVISC

Linear-Damper Properties

Defines the properties of a linear viscous damper.

Format and Example

1	2	3	4	5	6	7	8	9	10
PVISC	PID	C			PID	C			
PVISC	7	0.01							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
C	Damping constant.	R	0.0

Remarks

1. This entry defines a linear viscous damper. PVISC1can be used to define nonlinear dampers.
2. The behavior of this type of damper is discussed in *Dytran User's Guide*, Chapter 2: Elements, Lagrangian Elements

PVISC1

Nonlinear Damper Properties

Defines the properties of a nonlinear damper where the damping constant varies with the velocity.

Format and Example

1	2	3	4	5	6	7	8	9	10
PVISC1	PID	TABLE							
PVISC1	8	236							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
TABLE	Number of a TABLExx entry defining the variation of the force (y-value) with velocity (x-value).	I > 0	Required

Remarks

1. This entry defines the properties of a nonlinear damper. Use the PVISC entry to define linear dampers.
2. The values in the table are interpolated to get the force for a particular velocity.
3. The behavior of this damper is discussed in *Dytran User's Guide*, Chapter 2: Elements, Lagrangian Elements.

PVISCEX UDS Properties of a Linear Viscous Damper

Defines the properties of a User Defined Service of a linear viscous damper.

Format and Example

1	2	3	4	5	6	7	8	9	10
PVISCEX	PID	GROUP	PROP1	PROP2	PROP3	PROP4	PROP5	PROP6	
PVISCEX	100	USRVISC	4.0	1.4					

Field	Contents	Type	Default
PID	Property number	I > 0	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required
PROPi	Property values which can be used in the user subroutine	R	0.0

Remarks

1. The Property ID (PID) must be referenced by a CVISC entry.
2. See the explanation in Chapter 7: User Defined Services on how to use user-written subroutines.
3. The values for PROP(i) will be passed on the User Defined Services in variable PROP(1...6).

PWELD

Spotweld Property

Defines the properties of a spotweld connection between two grid points. It is referenced by the CROD or CBAR entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
PWELD	PID	FAILTENS	FAILCOMP	FAILSHEA	FAILTORQ	FAILBEND	FAILTOTF	FAILTOTM	+
PWELD	101	1.E5							+
+	FAILTIME								
+									

Field	Contents	Type	Default
PID	Property number	I > 0	Required
FAILTENS	Failure force in tension	R ≥ 0.0	No failure
FAILCOMP	Failure force in compression	R ≥ 0.0	No failure
FAILSHEA	Failure force in shear	R ≥ 0.0	No failure
FAILTORQ	Failure torque	R ≥ 0.0	No failure
FAILBEND	Failure bending moment	R ≥ 0.0	No failure
FAILTOTF	Failure total force	R ≥ 0.0	No failure
FAILTOTM	Failure total moment	R ≥ 0.0	No failure
FAILTIME	Failure based on time	R ≥ 0.0	No failure

Remarks

1. A spotweld is treated as a rigid body with its inertia properties calculated by lumping the properties of the end points.

   A set of spotwelds and/or BJOINs connected to each other is treated as one rigid body.

   Lumping of the initial positions and velocities:

   The lumped rigid-body mass is not zero:

   - The initial positions and velocities are lumped using mass-weighting.
   - If a grid point has zero mass, it's initial position and velocity is ignored.

   The lumped rigid-body mass is zero:

   - The initial positions and velocities are lumped by averaging.
   - Boundary conditions allocated to the grid points are combined, if possible.

When failure of a spotweld that is connected to other spotweld(s) and/or BJOINs occurs, the rigid-body lumped properties and boundary conditions are redefined.

2. If the end points of a spotweld coincide, the direction vector cannot be determined. As a result, no components of tension, compression, shear, torque, and bending can be calculated. Instead, the total force or moment is used to check for failure against the specified failure criteria:
   a. The total force acting on the spotweld is checked against:

      FAILTENS
      FAILCOMP
      FAILSHEA
      FAILTOTF

   b. The total moment acting on the spotweld is checked against:

      FAILTORQ
      FAILBEND
      FAILTOTM

   The spotweld fails if one of the above criteria is satisfied.
3. All failure modes are checked simultaneously.
4. An overview of the generated spotwelds and BJOINs can be requested. See PARAM,INFO-BJOIN.
5. You have access to the results of the spotweld elements by requesting for results of the corresponding CROD or CBAR elements.

   The variables are only calculated for spotwelds with a failure criterion. They are described as follows:

FAIL	Failure time	
XFORCE	Tension/compression force in the spotweld	
YFORCE	CROD	Shear force in the spotweld in direction of shear vector at end point 1
	CBAR	Shear force in the spotweld in the local y-direction, see CBAR for sign convention
ZFORCE	CROD	Shear force in the spotweld in direction of shear vector at end point 2
	CBAR	Shear force in the spotweld in the local z-direction, see CBAR for sign convention
XMOMENT	Torque in the spotweld	
YMOMENT	Bending moment in the spotweld in direction of bending moment vector at end point 1	
ZMOMENT	Bending moment in the spotweld in direction of bending moment vector at end point 2	
FIBL1	Mode of failure:	
	0	Not failed
	1	Failed on TAILTENS

	2	Failed on FAILCOMP
	3	Failed on FAILSHEA
	4	Failed on FAILTORQ
	5	Failed on FAILBEND
	6	Failed on FAILTOTF
	7	Failed on FAILTOTM
	8	Failed on FAILTIME

PWELD1

Skin-Stringer Delamination

Defines the properties of a skin-stringer connection with delamination or rupture criteria. The PWELD1 entry is referenced by a CROD entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
PWELD1	PID	FAILTENSL	FAILCOMPL	FAILSHEAL	FAILTORQL	FAILBENDL	FAILTOTFL	FAILTOTML	+
PWELD1	101	20.0							+
+	FAILTIME	POSITION							
+		UPPER							

Field	Contents		Type	Default
PID	Property number		I > 0	Required
FAILTENSL	Tensile failure force per unit length		R ≥ 0.0	No failure
FAILCOMPL	Compressive failure force per unit length		R ≥ 0.0	No failure
FAILSHEAL	Shear failure force per unit length		R ≥ 0.0	No failure
FAILTORQL	Torque failure moment per unit length		R ≥ 0.0	No failure
FAILBENDL	Bending failure moment per unit length		R ≥ 0.0	No failure
FAILTOTFL	Total failure force per unit length		R ≥ 0.0	No failure
FAILTOTML	Total failure moment per unit length		R ≥ 0.0	No failure
FAILTIME	Failure based on time		R ≥ 0.0	No failure
POSITION	Position of the stringer with respect to the skin element it is connected to:		C	MID
	MID	Stringer and skin are at the same location.		
	UPPER	Stringer is located on the upper side of the skin.		
	LOWER	Stringer is located on the lower side of the skin.		

Remarks

1. Connecting beam and shell grid points by a CROD element that references a PWELD1 entry defines a spotweld connection. The PWELD1 entry defines the failure criteria for the spotweld connection.
2. The spotweld connection is treated as a rigid body with the properties calculated by lumping the properties of the end points it connects.

   When the lumped rigid-body mass is not equal to zero:

   - The initial positions and velocities are lumped using mass weighting.
   - If a grid point has a zero mass, its initial position and velocity are ignored.

When the lumped rigid-body mass is equal to zero:

- The initial positions and velocities are lumped by averaging.
- Boundary conditions defined for the grid points of the connection are combined when possible.

3. The end point a of the CROD is the connecting point of the skin (shell element), and end point 2 is the connecting point of the stringer (beam element).
4. The failure force and/or moment criteria are defined per unit length, where the length is defined by the length of all beam (stringer) elements connected to the spotweld element. Each stringer contributes half of its length to the spotweld. The resulting forces and moments per unit length, acting on the spotweld are checked against the failure criteria as defined on the PWELD1 entry.
5. The position of the beam (stringer) element with respect to the shell (skin) element it is connected to, should be defined by considering the orientation of the element's local y-z coordinate axes. The normal to the skin coincides with the z_s axis. The upper side of the skin is defined by the direction the normal points to. Figure 5-13 illustrates the definition of the upper- and the lower-side positioning of the connection.

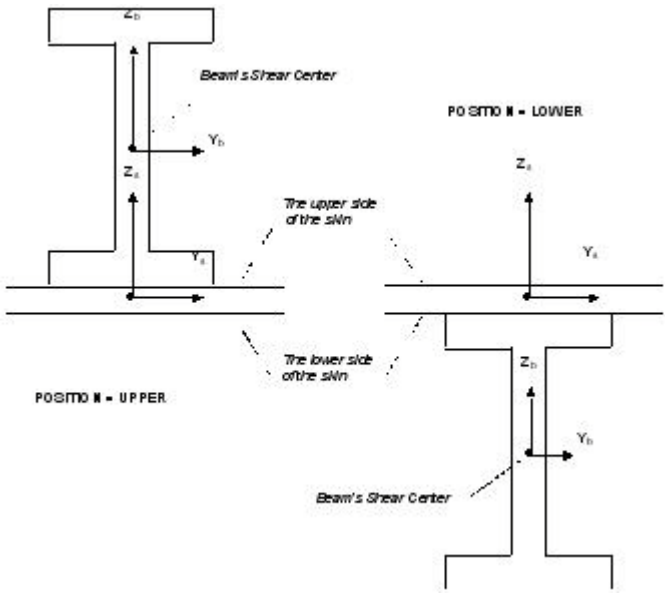


Figure 5-13 Definition of the Upper- and the Lower-side Positioning

6. The direction of the normal at the location of the spotweld connection is defined as the average of the normal vectors of all connected shell (skin) elements. The average direction is used in the calculation of the failure criteria. The length of the spotweld element is small and assumed to be zero. As a result, the forces acting at the end points of the spotweld do not introduce additional moments.

7. In case the position is defined as MID, the direction of the spotweld element cannot be determined. As a result, the tensile, compressive and shear force components, as well as the torque and bending moment components cannot be computed. The failure criteria is based on the total force and/or moment. Note that the output for the force components is the total force, and the output for the moment components is the total moment.
8. The total force acting on the spotweld is checked against:

   FAILTENSL

   FAILCOMPL

   FAILSHEAL

   FAILTOTL

   The total moment acting on the spotweld is checked against:

   FAILTORQL

   FAILBENDL

   FAILTOTML

   The spotweld connection fails if any of the above criteria is met.
9. All failure criteria are checked simultaneously. If any of the failure criteria is met, the connection will fail.
10. Tensile and compressive loading is evaluated in the direction of the normal to the surface. The shear load is evaluated in the plane of the surface. A positive load on the spotweld element in the direction of the normal (the XFORCE) indicates tensile loading. A negative load indicates compressive loading. Note that you input both the tensile and compressive failure criteria as positive numbers.
11. The shear load is evaluated in the plane to the normal. The output value on the connecting element is always positive, as the direction of the shear in the plane is irrelevant for failure.
12. The torque is evaluated as the moment about the normal to the surface. The bending moment is evaluated in the plane to the normal. The output of the moments on the connecting element is always positive, as the direction of the moments is irrelevant for failure.
13. An overview of the generated spotwelds can be requested by the entry PARAM,INFO-BJOIN. See the reference page for more details.
14. You can access the results of the spotweld elements for output by requesting results for the corresponding CROD elements. The variables listed below are available only for CROD elements that have been used to define a spotweld element:

XFORCE	Tensile/compressive force in the spotweld
YFORCE	Shear force in the direction of the shear vector at end point 1
ZFORCE	Shear force in the direction of the shear vector at end point 2
XMOMENT	Torque in the spotweld
YMOMENT	Bending moment in the spotweld in the direction of the bending moment vector at end point 1

ZMOMENT	Bending moment in the spotweld in the direction of the bending moment vector at end point 2	
FIBL1	Failure mode:	
	0	Not failed
	1	Failed on tension
	2	Failed on compression
	3	Failed on shear
	4	Failed on torque
	5	Failed on bending
	6	Failed on total force
	7	Failed on total moment

15. When you have not defined any failure criteria, the output results on the CROD elements that define the connection are all zero. The failure evaluation and computation is skipped in this case for performance reasons. If you want to see the actual load on the connecting elements, but do no wish to introduce any failure, you have to define at least one criterion with a large enough value to ensure there is no failure. Any value less than 1.0E20 suffices.
16. Spotwelds for a skin-stringer connection can also be defined using a BJOIN entry with the TYPE set to RUPTURE. Note that in case you use the BJOIN option, you do not have access to the results on the connection as you have when using the PWELD1 definition. The BJOINconnection does not use the CRODelements as the connecting entities and therefore no output on the connection is available.

PWELD2

Sandwich Structure Delamination

Defines the failure properties for delamination/peeling of the facing and core connection of a sandwich structure. The PWELD2 entry is referenced by a CROD entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
PWELD2	PID	FAILTENSA	FAILCOMPA	FAILSHEA	FAILTORQA	FAILBENDA	FAILTOTFA	FAILTOTMA	+
PWELD2	101	1.0							+
+	FAILTIME								
+									

Field	Contents	Type	Default
PID	Property number	I > 0	Required
FAILTENSA	Tensile failure force per unit area	R ≥ 0.0	No failure
FAILCOMPA	Compressive failure force per unit area	R ≥ 0.0	No failure
FAILSHEAA	Shear failure force per unit area	R ≥ 0.0	No failure
FAILTORQA	Torque failure moment per unit area	R ≥ 0.0	No failure
FAILBENDA	Bending failure moment per unit area	R ≥ 0.0	No failure
FAILTOTFA	Total failure force per unit area	R ≥ 0.0	No failure
FAILTOTMA	Total failure moment per unit area	R ≥ 0.0	No failure
FAILTIME	Failure based on time	R ≥ 0.0	No failure

Remarks

1. The PWELD2 connection can only be used for a shell (facing) to solid (core) element connection.
2. The sandwich structure is modeled using shell and solid elements. The connection of the facing and the core is modeled by connecting the shell and solid element's grid points with CROD elements. The CROD elements should refer to a PWELD2 entry.
3. When the facing-core connection has not yet failed, it is treated as a rigid body with the properties calculated by lumping the properties of the end points it connects.

   When the lumped rigid-body mass is not equal to zero:

   - The initial positions and velocities are lumped using mass weighting.
   - If a grid point has a zero mass, its initial position and velocity are ignored.

   When the lumped rigid-body mass is equal to zero:

   - The initial positions and velocities are lumped by averaging.
   - Boundary conditions defined for the grid points of the connection are combined when possible.

4. End point 1 of the CROD is the connecting point of the facing (shell element) and end point 2 is the connecting point of the core (solid element).
5. The failure criteria are based on forces and moments per unit area. The area associated with a connection is equal to the sum of the areas of the connected shell (facing) elements. Each facing elements contributes one quarter of its area to the weld connection in case of a quad element (CQUAD4), and one third in case of a triangular element (CTRIA3). The resulting forces and moments are checked against the failure criteria defined on the PWELD2 entry.
6. All failure criteria are checked simultaneously. If any of the failure criteria is met, the connection fails.
7. Since the solid element grid points have only three degrees of freedom (translation only), there is no contribution of moments or rotations to the spotweld element from the solid's grid points. The contribution of the moment and rotation comes entirely from the connected shell elements.
8. The face of a solid element that is connected to the shell (facing) element determines the direction of the normal to the surface. The normal to the surface always points outwards. The normal is used as the direction vector in the computation of the failure criteria for compression and tension. The direction of the normal is defined as the average of the normal vectors of the connected core element's faces.
9. Tensile and compressive loading is evaluated in the direction of the normal to the surface. The shear load is evaluated in the plane to the normal. A positive load on the spotweld element in the direction of the normal (the XFORCE) indicates tensile loading. A negative load indicates compressive loading. Note that you input both the tensile and compressive failure criteria as positive numbers.
10. The shear load is evaluated in the plane to the normal. The output of the shear load on the connecting element is always positive, as the direction of the shear in the plane is irrelevant for failure.
11. The torque is evaluated as the moment about the normal to the surface. The bending moment is evaluated in the plane to the normal. The output of the moments on the connecting element is always positive, as the direction of the moments is irrelevant for failure.
12. An overview of the generated connections can be requested by the entry PARAM,INFO-BJOIN. For more details, see the parameter BJOIN reference page.
13. You can access the results of the spotweld elements for output by requesting results for the corresponding CROD elements. The variables listed below are available only for CROD elements that have been used to define a spotweld element:

XFORCE	Tensile/compressive force in the spotweld
YFORCE	Shear force in the direction of the shear vector at the facing
ZFORCE	Shear force in the direction of the shear vector at the core
XMOMENT	Torque in the spotweld
YMOMENT	Bending moment in the spotweld in the direction of the bending moment vector at the facing
ZMOMENT	Bending moment in the spotweld in the direction of the bending moment vector at the core

FIBL1	Failure mode:	
	0	Not failed
	1	Failed on tension
	2	Failed on compression
	3	Failed on shear
	4	Failed on torque
	5	Failed on bending
	6	Failed on total force
	7	Failed on total moment

14. When you have not defined any failure criteria, the output results on the CROD elements that define the connection will all be zero. The failure evaluation computations are skipped in this case for performance reasons. If you want to see the actual load on the connecting elements, but do not wish to introduce any failure, you have to define at least one criterion with a large enough value to ensure there is no failure. Any value less than 1.0E20 suffices.

RBC3

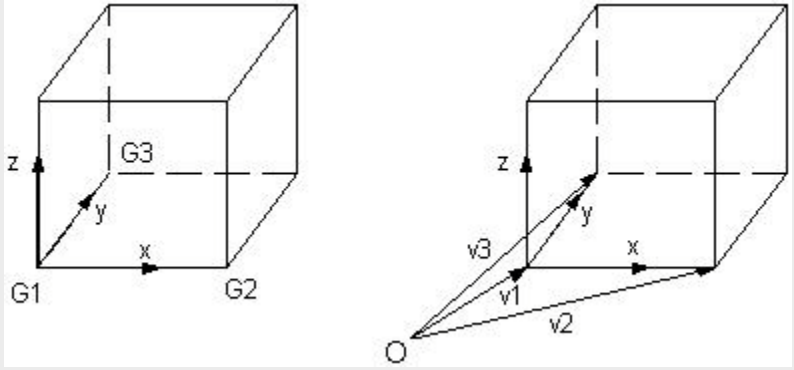
Rigid-Body Constraint

Defines a three-point constraint on a RIGID surface, a MATRIG, or RBE2-FULLRIG rigid body.

Format and Example

1	2	3	4	5	6	7	8	9	10
RBC3	RID	MID	C	G1	G2	G3			+
RBC3	3	MR5	12	26	23	27			+
+	X1	Y1	Z1	X2	Y2	Z2			+
+									+
+	X3	Y3	Z3						
+									

Field	Contents	Type	Default
RID	Unique rigid-body constraint number	I > 0	Required
MID	Number, MR<Number>, or FR<Number>, where a number refers to a RIGID surface, MR<Number> refers to a MATRIG, and FR<Number> refers to an RBE2-FULLRIG entry.	C or I	Required
C	Component number of local coordinate (any unique combination of the digits 1 through 6 with no embedded blanks).	I > 0	See Remark 3.
G1, G2, G3	Grid-point numbers defining the RBC3 coordinate system	I > 0	See Remark 1.
X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3	Coordinates of three points defining the RBC3 coordinate system shown below.	R	See Remark 1.



Remarks

1. If G1, G2, and G3 are specified, then the RBC3 coordinate system is determined by the grid points. The position vectors for G1, G2, and G3 are denoted by v1, v,2 and v3, respectively. If G1, G2, and G3 are not specified, then the coordinate system is either specified by the vectors v1 = (X1, Y1, Z1), v2 = (X2, Y2, Z2) and v3 = (X3, Y3, Z3) if X1 through Z3 are specified, or by the vectors v1 = (0, 0, 0), v2 (1, 0, 0), and v3 = (0, 1, 0), by default. The local x-axis is the normalized vector v2-v1. The local z-axis is the normalized cross product of the vectors v2-v1 and v3-v1 and is thus perpendicular to the plane spanned by these vectors. The local y-axis is the cross product of the local z- and x-axis.
2. The grid points G1, G2, and G3 must be unique. Also, the vectors (X1, Y1, Z1), (X2, Y2, Z2), and (X3, Y3, Z3) must be unique.
3. The translational and rotational constraints are applied to the center of gravity of the rigid body in the local coordinate system.

RBE2

Rigid-Body Element

Defines a set of grid points that form a rigid element.

Format and Example

1	2	3	4	5	6	7	8	9	10
RBE2	EID	G1	CM	G2	G3	G4	G5	G6	+
RBE2	9	8	12	10	12	14	15	16	+
+	G7	G8	THRU	G10	-etc.-				
+	20	25	THRU	32					

Field	Contents	Type	Default
EID	Number of the rigid-body element	I > 0	Required
G1..Gn	Grid-point numbers with degrees of freedom that are specified by CM are coupled.	I > 0	Required
CM	Component numbers of the grid points that are coupled. These are in the basic coordinate system. The components are indicated by any of the digits 1, 2, 3, 4, 5, or 6 with no embedded blanks. Combinations are allowed, e.g., 12, 123. In case the rigid element should behave as a full rigid body, CM should read FULLRIG.	See Remark 7.	Required

Remarks

1. The element number should be unique with respect to all other rigid element numbers.
2. The RBE2 definition allows particular degrees of freedom of a set of grid points to be coupled so that the grid points always move the same amount. The motion of the set of grid points is the weighted average of the motion of all the grid points for the degrees of freedom coupled through the RBE2 definition.
3. The component numbers refer to the basic coordinate system.
4. Loads, initial velocities, or constraints should be applied to the first independent grid point. They are then applied to the coupled degrees of freedom for all the grid points defined on the RBE2 entry.
5. Both rotational and translational degrees of freedom can be coupled.
6. Grid points associated with rigid surfaces cannot be part of an RBE2 grid point list.
7. Instead of defining coupled components, it is possible to define the RBE2entry as a single rigid body by using the FULLRIG option. The geometric properties of the rigid body are calculated from the geometry and the mass of the grid points.
8. Grid points referred to by the JOIN entry cannot be part of an RBE2 grid point list.

9. It is possible to merge an RBE2entry with a MATRIG entry by using the FULLRIG option and PARAMMATRMERG or PARAM,MATRMRG1. A normal RBE2 entry (with constraint) however cannot be merged with a MATRIG entry or an RBE2-FULLRIG entry.
10. By using PARAM,CFULLRIG, all 123456 constraints on a normal RBE2 are automatically converted to the FULLRIG option.
11. By using PARAM,RBE2INFO,GRIDON, the grid points of the RBE2are listed in the output file.
12. Lagrangian Elements for a description of the use of RBE2.

RBHINGE

Rigid Body Hinge

Defines a hinge between a rigid body and a deformable structure on the common six degrees of freedom.

Format and Example

1	2	3	4	5	6	7	8	9	10
RBHINGE	RID	MID	C	G1	G2	THRU	G3	BY	+
RBHINGE	1	14	456	1	10	THRU	100	BY	+
+	G5	-etc.-							
+	2								

Field	Contents	Type	Default
RID	Unique rigid body hinge number	I > 0	Required
MID	Number, MR<Number>, or FR<Number>, where a number refers to a RIGID surface, MR<Number> refers to a MATRIG, and FR<Number> refers to an RBE2-FULLRIG entry.	C or I	Required
C	Component number of rotation which is defined as a hinge (any unique combination of the digits 4, 5 or 6).	I > 0	456
Gi	Grid point numbers. THRU indicates a range of grid points. BY is the increment within this range.	I > 0	Required

Remarks

1. When grid points are part of a rigid body and a deformable structure, they transfer rotational momentum if they posses six degrees of freedom. This is the case when they are connected to a beam/bar, tria or quad shell element. Using the RBHINGE entry specifies rotational degrees of freedom that can be defined as behaving as a hinge.

   RBHINGE is not needed for grid points that have only three degrees of freedom, since the hinge is defined by default.

2. The component number refers to the global coordinate system.

RCONN

Rigid Connection

Defines a rigid connection between the different parts of Lagrangian meshes (tied surfaces).

Format and Example

1	2	3	4	5	6	7	8	9	10
RCONN	CID	STYPE	PTYPE	SID	PID	OPTION			+
RCONN	7	GRID	SURF	3	7	NORMAL			+
+									+
+									+
+	CLSGAP	GAPDIS	GAPDISV						
+									

Field	Contents		Type	Default
CID	Unique rigid-connection number		I > 0	Required
STYPE	Type of entity used to define the secondary contact surface		C	SURF
	SURF	A SURFACE entry is used to select the faces of the elements on the secondary contact surface. SID is the number of the SURFACE entry. See Remark 2.		
	GRID	Grid points are tied to the primary contact surface. SID then refers to a SET1entry containing the list of grid points to be used. See Remarks 3. and 4.		
PTYPE	Type of entity used to define the primary contact surface		C	SURF
	SURF	A SURFACE entry is used to select the faces of the elements on the primary contact surface. PID is the number of the SURFACE entry.		
SID	The number of a secondary contact SURFACE entry or the number of a SET1 entry containing the list of grid points		I > 0	Required
PID	The number of a primary contact SURFACE entry		I > 0	Required
OPTION	Only used if discrete grid points are tied to a surface (STYPE is equal to GRID).		C	NORMAL
	NORMAL	The grid points are tied to the primary contact surface. See Remark 3.		
	SHELL	The grid points are attached to the edge of shell or beam elements, which are tied to the shell surface. See Remark 4.		

Field	Contents		Type	Default
CLSGAP	Switch to automatically close any gaps that are present between the primary-secondary contact surface.		C	No
	YES	Gaps are automatically closed.		
	NO	Gaps are not closed. See Remark 6.		
GAPDIS	Defines the tolerance used in the search for a primary contact face. If the distance between a secondary contact point and a primary contact face falls within this tolerance, the primary contact face is accepted. If not, the search for a correct primary contact face continues.		C	DISTANCE
	FACTOR	The tolerance has the length of: (GAPDISV) * (Minimum side of faces in secondary contact surface). See Remark 9.		
	DISTANCE	The tolerance has the length as specified at GAPDISV.		
GAPDISV	The value of the gap tolerance or a factor to calculate this tolerance depending on the value of GAPDIS.		R	1.E20

Remarks

1. The RCONN entry can be used to define three types of connection as described in *Dytran Theory Manual*, Chapter 3: Constraints and Loading, Lagrangian Elements.
2. Two Surfaces Tied Together

   Define secondary contact and primary contact segments representing the two surfaces to be tied together. A small gap between the two sets of segments is only allowed when the two meshes are not identical. The two surfaces are tied together during the analysis.
3. Grid Points Tied to a Surface

   If STYPE is set to GRID and OPTION is set to NORMAL, the secondary contact entities comprise discrete grid points that are tied to the primary contact surface during the analysis. The grid points must lie on the surface.
4. Shell Edge Tied to a Shell Surface

   If STYPE is set to GRID and OPTION is set to SHELL, the edges of shell or beams elements can be tied to the faces of other shells. The grid points attached to the edge of the shells/beams must be selected as the secondary contact grid points. The shell surface to which they are tied must be selected as the primary contact surface. The two sets will then be tied together throughout the analysis. All degrees of freedom will be coupled.
5. The CLSGAP entry enables you to define two different meshes that are not coincident over the primary/secondary contact interface. If the option is set to YES, the secondary contact surface becomes coincident (according to projections) with the primary contact surface.

6. The search method of the contact algorithm is used to find the closest primary contact face. The tolerance defined with the GAPDIS/GAPDISV fields is similar to the monitoring distance defined on the CONTACT entry with the MONDIS/MONDISV fields.
7. The use of the gap closing CLSGAP can cause an element to collapse. This may happen if the GAPDISV tolerance is set to a value greater than the length of the side of an element.
8. When a solid and a shell mesh are tied together, the rotational degrees of freedom of the shell grid points are not coupled.
9. If STYPE is set to GRID, the option FACTOR in the GAPDIS field is not allowed.
10. Avoid the following situation when using the RCONN entry:



RCONN1: surface 1 as secondary contact of surface 2

RCONN2: surface 1 as secondary contact of surface 3

In this situation, the corner point of surface 1 has two primary contact surfaces to follow. Therefore, the mass and the force of the corner point are lumped twice.

RCONREL

Rigid Connection with Rigid Ellipsoids

Defines a connection between a rigid ellipsoid and Lagrangian grid points or rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
RCONREL	RID	SIDC	TYPE	SID					
RCONREL	20	30	GRID	40					

Field	Contents		Type	Default
RID	Unique number of an RCONREL entry		I > 0	Required
SIDC	Number of a SETC entry giving the name of the rigid ellipsoid to which entities are connected. See Remark.		I > 0	Required
TYPE	The type of entities that are connected to the rigid ellipsoid.		C	Required
	GRID	Grid points.		
	RIGID	Rigid surface, RBE2-FULLRIG, and MATRIG.		
SID	The number of a SET1 entry listing the grid points or rigid surfaces that are connected to the rigid ellipsoid. In case a MATRIG or an RBE2-FULLRIG entry is connected to the rigid ellipsoid, SID refers to a SET1 entry listing MR<id> or FR<id>, where id refers to a MATRIG or an RBE2-FULLRIG entry, respectively.		I > 0	Required

Remark

The SETC entry can only contain the name of one ellipsoid.

RELEX External Definition of a Rigid Ellipsoid

Defines a rigid ellipsoid whose properties and motion are defined by either ATB.

Format and Example

1	2	3	4	5	6	7	8	9	10
RELEX	NAME	PROG							
RELEX	HEAD								

Field	Contents		Type	Default
NAME	This name is used within the Dytran input file to define the interactions between the external ellipsoid and Dytran grid points and rigid bodies. This name is also used in the output requests.		C	Required
	When coupled to ATB, the name must correspond to the name of the ATB segment.			
PROG	Name of the external program:		C	Required
	ATB	Dytran runs coupled with ATB	C	Required

Remarks

1. This entry should only be used when Dytran is used with ATB.
2. Rigid ellipsoids can be defined directly within Dytran using the RELLIPS entry.
3. RELEX and RELLIPS entries can not be mixed in the same model.
4. For ATB, only the segment contact ellipsoid can be used. The name of the contact ellipsoid is equal to the name of the segment, as specified on the first field of the B.2 entry in the ATB input file.
5. See *Dytran User's Guide*, Chapter 7: Interface to Other Applications, ATB Occupant Modeling Program for instructions on how to use ATB.

RELLIPS

Rigid Ellipsoid

Defines an analytical rigid ellipsoid.

Format and Example

1	2	3	4	5	6	7	8	9	10
RELLIPS	NAME	A	B	C	MASS	XCG	YCG	ZCG	+
RELLIPS	10	0.1	10.0	10.0	0.1	0.	0.	0.	+
+	XL	YL	ZL	XS	YS	ZS			+
+	0.	0.	1.	1.	0.	0.			+
+	VX	VY	VZ	WA	WB	WC			
+	-0.1								

Field	Contents	Type	Default
NAME	Ellipsoid name	C	Required
A, B, C	Size of the ellipsoid in the a-, b-, and c-directions (a > b > c)	R > 0.0	Required
MASS	Mass of the ellipsoid	R > 0.0	Required
XCG, YCG, ZCG	Coordinates of the geometric center of the ellipsoid (the geometric center of the ellipsoid coincides with the center of gravity)	R	0.0
XL, YL, ZL	Vector defining the orientation of the longest axis of the ellipsoid	R	0.0
XS, YS, ZS	Vector defining the orientation of the shortest axis of the ellipsoid	R	0.0
VX, VY, VZ	Initial translational velocities of the center of the ellipsoid in the x-, y-, and z-directions	R	0.0
WA, WB, WC	Initial rotational velocities of the ellipsoid in the a-, b-, and c-directions.	R	0.0

Remark

RELEX and RELLIPS entries cannot be mixed in the same model.

RFORCE

Rotational Force Field

Defines loading due to a centrifugal acceleration field.

Format and Example

1	2	3	4	5	6	7	8	9	10
RFORCE	LID	G		SCALE	NX	NY	NZ		
RFORCE	29	2		37.6	1.0	2.0	0.		

Field	Contents	Type	Default
LID	Number of a set of loads	I > 0	Required
G	Grid-point number on the axis of rotation.	I > 0	Required
SCALE	Scale factor for rotational velocity. See Remark 6.	R	1.0
NX, NY, NZ	Components of the rotational-direction vector. At least one component must be nonzero. The vector (NX, NY, NZ) acts at grid point G.	R	0.

Remarks

1. The rotational velocity is calculated as

   $$W(t) = T(t)*SCALE*N$$

   where $SCALE$ is the scale factor, N the directional vector (NX, NT, NZ), and $T(t)$ the value at time t interpolated from the table or function referenced by the TLOADn entry.
2. LID must be referenced by a TLOADn entry.
3. The type field on the TLOADn entry must be set to zero.
4. Only one centrifugal force field can be defined in the problem.
5. Centrifugal forces act on all Lagrangian structural elements and rigid surfaces.
6. The rotation is input in revolutions per unit time.
7. The X, Y, and Z location of node G is used to determine the axis of rotation and this remains constant in time. Subsequently, if the node G moves during the simulation, the axis of rotation will not move with it, but remain stationary.

RIGID

Rigid Surface

Defines a rigid surface.

Format and Example

1	2	3	4	5	6	7	8	9	10
RIGID	RID	SID	MASS		XCG	YCG	ZCG		+
RIGID	25	32	527.		117.6	339.4	21.0		+
+		VX	VY	VZ	CID	WX	WY	WZ	+
+									+
+		IXX	IXY	IXZ	IYY	IYZ	IZZ		+
+		4495.			4495.		4495.		+
+	VELMAX	DSPMAXN	DSPMAXP	XDIR	YDIR	ZDIR			
+			10.0	0.0	0.0	1.0			

Type	Contents	Type	Default
RID	Unique rigid-surface number	I > 0	Required
SID	Number of a SURFACE entry defining the shape of the rigid surface	I > 0	Required
MASS	Mass of the rigid body	R > 0.0	Required
XCG, YCG, ZCG	Coordinates of the center of gravity of the rigid body	R	Required
VX, VY, VZ	Initial translational velocities of the center of mass in the basic coordinate system	R	0.0
CID	Number of a CORD2C entry	I	0
WX, WY, WZ	Initial rotational velocities, relative to a coordinate system with its origin at the center of gravity, and its axes aligned with the coordinate system CID.	R	0.0
IXX, IXY, IXZ IYY, IYZ, IZZ	Moments of inertia, relative to a coordinate system with its origin at the center of gravity, and its axes aligned with the coordinate system CID	R	See Remark 2.
VELMAX	Limiting the maximum magnitude of the translational velocity.	R≥0.0	See Remark 4.
DSPMAXN	Limiting the maximum of the translational displacement in the negative direction of the rigid body control.	R≥0.0	See Remark 4.

Type	Contents	Type	Default
DSPMAXP	Limiting the maximum of the translational displacement in the positive direction of the rigid body control.	R≥0.0	See Remark 4.
XDIR, YDIR, ZDIR	Vector indicating the positive direction of the translational rigid body control.	R	See Remark 4.

Remarks

1. A CID of zero references the basic coordinate system.
2. The default for IXX, IYY, and IZZ is 1.E10; the default for IXY, IXZ, and IYZ is zero.
3. The mass of the rigid surface is distributed to the grid points on the surface.
4. The maximum translational motion of a rigid body can be restricted by the optional VELMAX, DSPMAXN, and DSPMAXP entries. Blank entries correspond to no motion control, while 0.0 entries set the limiting displacement and velocity values to 0.0. DSPMAXN and DSPMAXP limit the maximum displacement of a rigid body in positive and negative directions of a local axis defined by (XDIR, YDIR, ZDIR). If either DSPMAXN or DSPMAXP is specified, then the (XDIR, YDIR, ZDIR) must be defined. This feature is designed primarily to control the motion of rigid tooling in metal forming applications, so it must be used with caution in other scenarios. For constraining rigid body motion in general applications, the users are encouraged to use RBC3 or TLOAD1 (TYPE=12) instead.

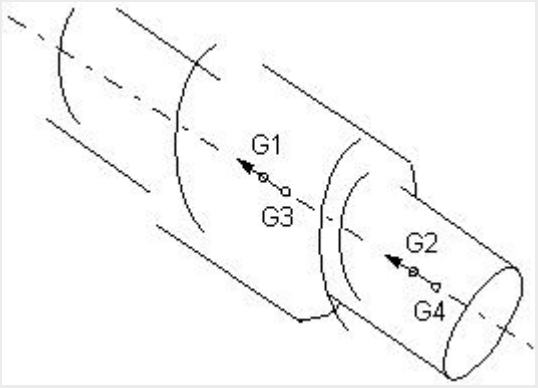
RJCYL Cylindrical-Joint Constraint Between Rigid Bodies

Defines a cylindrical joint between grid points on two rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
RJCYL	ID	STIFF	G1	G2	G3	G4			
RJCYL	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number	I > 0	Required
STIFF	Relative stiffness of the joint	R	1.0
G1-G4	Grid-point numbers defining the joint connectivity	I > 0	Required



Remarks

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body.
3. The vector from G1 to G3 determines the axis of sliding. Spring forces are calculated between G1 and G2 and between G3 and G4 to keep all four points on the axis of sliding.
4. If the initial position of grid points G2 and/or G4 is off the axis of sliding a force in the joint is initialized.
5. The absolute stiffness of the rigid body joints is calculated automatically by Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

6. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM, RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.
7. Although the joint is designed for usage with rigid bodies, it is allowed to use finite-element grid points.
8. RJCYL can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

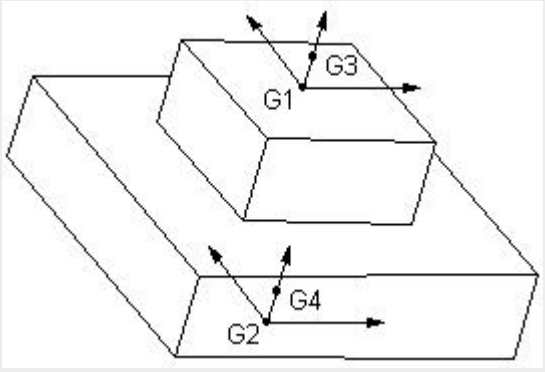
RJPLA

Planar-Joint Constraint Between Rigid Bodies

Defines a planar joint between grid points on two rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
RJPLA	ID	STIFF	G1	G2	G3	G4			
RJPLA	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number	I > 0	Required
STIFF	Relative stiffness of the joint	R	1.0
G1-G4	Grid-point numbers defining the joint connectivity	I > 0	Required
			

Remarks

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body.
3. The vector from G1 to G3 defines the normal to the plane on which the two bodies can slide relative to each other. G2 should lie on the plane through G1. Spring forces are calculated between G1 and G2 and between G3 and G4 to keep all four points in the plane of sliding.
4. If the initial position of grid points G2 and/or G4 is off the normal to the plane of sliding, a force in the joint is initialized.
5. The absolute stiffness of the rigid-body joints is calculated automatically by Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

6. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM, RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.
7. Although the joint is designed for usage with rigid bodies, it is allowable to use finite-element grid points.
8. RJPLA can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2FULLRIG entries.

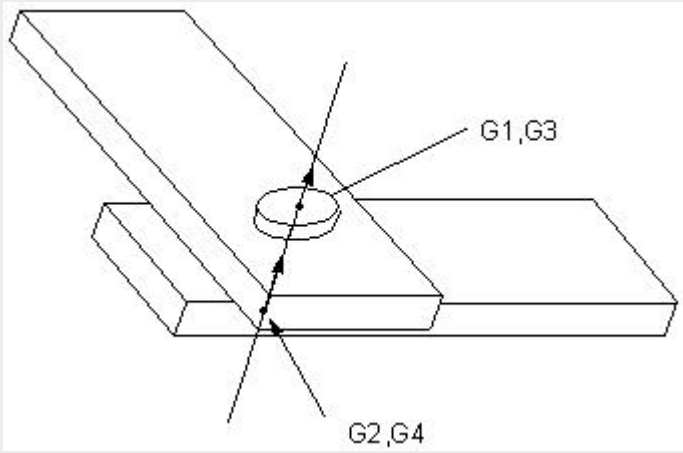
RJREV — Revolute-Joint Constraint Between Rigid Bodies

Defines a revolute joint (hinge) between grid points on two rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
RJREV	ID	STIFF	G1	G2	G3	G4			
RJREV	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number	I > 0	Required
STIFF	Relative stiffness of the joint	R	1.0
G1-G4	Grid point numbers defining the joint connectivity	I > 0	Required



Remarks

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body. G1 and G2 should be coincident, and G3 and G4 should be coincident.
3. The vector from G1 to G3 determines the axis about which the two bodies can rotate. Spring forces are calculated between G1 and G2 and between G3 and G4 to keep all four points on the axis of rotation.
4. The absolute stiffness of the rigid-body joints is calculated automatically by Dytran. The stiffness of the joints is calculated so that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

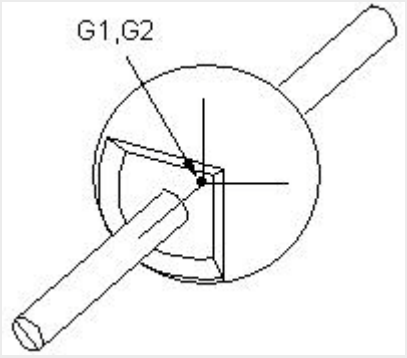
5. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM, RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.
6. Although the joint is designed for usage with rigid bodies, it is allowed to use finite-element grid points.
7. RJREV can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

RJSPH Spherical-Joint Constraint Between Rigid Bodies

Defines a spherical (ball) joint between grid points on two rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
RJSPH	ID	STIFF	G1	G2					
RJSPH	9	1.0	47	173					

Field	Contents	Type	Default
ID	Unique joint number	I > 0	Required
STIFF	Relative stiffness of the joint	R	1.0
G1-G2	Grid-point numbers defining the joint connectivity	I > 0	Required
			

Remarks

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 belongs to the first rigid body, G2 belongs to the second rigid body. G1 and G2 should be coincident. Spring forces are calculated between G1 and G2 so that the two bodies can rotate about the joint.
3. The absolute stiffness of the rigid-body joints is calculated automatically by Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.
4. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM, RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

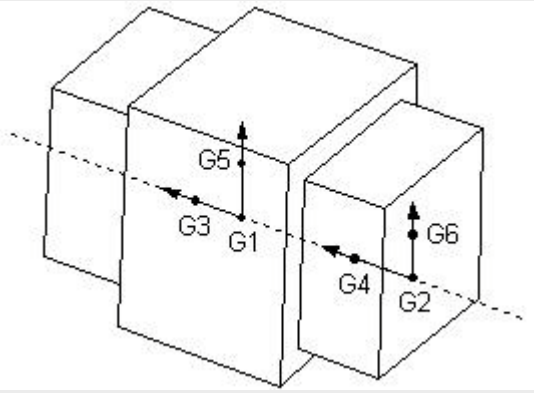
5. Although the joint is designed for usage with rigid bodies, it is allowable to use finite-element grid points.
6. RJSPH can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by theMATRIG or RBE2-FULLRIG entries.

RJTRA Translational-Joint Constraint Between Rigid Bodies

Defines a translational joint, which allows relative sliding but no rotation, between grid points on two rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
RJTRA	ID	STIFF	G1	G2	G3	G4	G5	G6	
RJTRA	9	1.0	47	173	53	269	17	87	

Field	Contents	Type	Default
ID	Unique joint number	I > 0	Required
STIFF	Relative stiffness of the joint	R	1.0
G1-G6	Grid-point numbers defining the joint connectivity	I > 0	Required
			

Remarks

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1, G3, and G5 are grid points belonging to the first rigid body; G2, G4, and G6 are points belonging to the second rigid body.
3. The vector from G1 to G3 determines the axis along which the two bodies can slide relative to each other. The vectors from G1 to G5 and from G2 to G6 are perpendicular to the axis of sliding. Spring forces are calculated between G1 and G2, between G3 and G4, and between G5 and G6 to keep the first four grid points on the axis of sliding and the other two grid points on a vector that is parallel to the axis of sliding.
4. If the initial position of grid points G2 and/or G4 is off the axis of sliding a force in the joint is initialized. If the initial vector from G5 to G6 is not parallel to the vector from G1 to G3, a force in the joint is initialized.

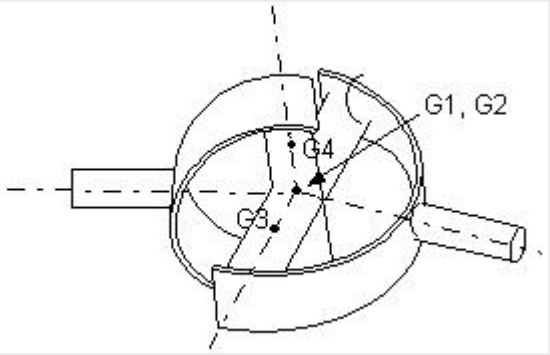
5. The absolute stiffness of the rigid-body joints is calculated automatically by Dytran. The stiffness of the joints is calculated so that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.
6. The absolute stiffness of the rigid body joints is multiplied by a factor defined on `PARAM, RJSTIFF`. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.
7. The grid points used in the definition of the joint do not have to be rigid-body joints, but may also be finite-element grid points.
8. `RJTRA` can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIGor RBE2`-FULLRIG` entries.

RJUNI — Universal-Joint Constraint Between Rigid Bodies

Defines a universal joint between grid points on two rigid bodies.

Format and Example

1	2	3	4	5	6	7	8	9	10
RJUN	ID	STIFF	G1	G2	G3	G4			
RJUN	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number	I > 0	Required
STIFF	Relative stiffness of the joint	R	1.0
G1-G4	Grid-point numbers defining the joint connectivity	I > 0	Required
			

Remarks

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body. G1 and G2 should be coincident, while G3 and G4 cannot be coincident.
3. G3 and G4 define the orientation of the rotation of the joint, as shown in the figure above. Spring forces are calculated between G1 and G2 as in the spherical joint and between G3 and G4, based on the Pythagorean theorem.
4. The absolute stiffness of the rigid-body joints is calculated automatically by Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.
5. The absolute stiffness of the rigid body joints is multiplied by a factor defined on PARAM, RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

6. Although the joint is designed for usage with rigid bodies, it is allowable to use finite-element grid points.
7. RJUNI can be applied to rigid bodies defined by the RIGIDentry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

RUBBER1 Mooney-Rivlin Rubber Material

Defines a nearly incompressible hyperelastic material for Lagrangian solid elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
RUBBER1	MID	RHO	A	B	NU				+
RUBBER1	3	1000.	0.34	0.27	0.495				+
+			BULKTYP	BULKQ	BULKL				
+									

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
A	Strain-energy density function constant		R	Required
B	Strain-energy density function constant		R	Required
NU	Poisson's ratio		0.0 ≤ R < 0.5	Required
BULKTYP	Bulk-viscosity model		C	DYNA
	DYNA	Standard DYNA3D model		
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient		R ≥ 0.0	0.0

Remarks

1. The continuation line with the bulk-viscosity data can be omitted.
2. The constitutive behavior of this material is defined as a total stress/total strain relationship. The nonlinear elastic material response is formulated by a strain-energy density function for large-strain components rather than by Hooke's law.

   The strain-energy density function is formulated according to the Mooney-Rivlin model and is defined as

$$W(I_1,I_2,I_3) = A(I_1-3)+B(I_2-3)+C\left[\frac{1}{I_3^2}-1\right]+D(I_3-1)^2$$

   The constants C and D are defined as:

$$C = \frac{1}{2}A + B \quad D = \frac{A(5v-2)+B(11v-5)}{2(1-2v)}$$

where A, B, and v are input parameters.

I_1, I_2, and I_3 are strain invariants in terms of stretches.

For a rubber-like material, the shear modulus G is much less than the bulk modulus K. As a result, Poisson's ratio v is nearly equal to one half.

3. This material can only be used with Lagrangian solid elements.
4. The behavior of this material is discussed in more detail in *Dytran Theory Manual*, Chapter 3: Materials.
5. When the elements with this material deform severely, it is possible that the job will be unstable. In this case, please use PARAM,HGTYPE,DYNA instead of default hourglass control.

RUBBER2 Ogden Rubber Material

Defines a nearly incompressible hyperelastic material using Ogden function for Lagrangian solid elements.

Format and Example

1	2	3	4	5	6	7	8	9	10
RUBBER2	MID	RHO	NU						
RUBBER2	3	1000.	0.495						

+	MU1	MU2	MU3	MU4	MU5	MU6	MU7	MU8	
+	6.30	0.12	-0.10						

+	ALPHA1	ALPHA2	ALPHA3	ALPHA4	ALPHA5	ALPHA6	ALPHA7	ALPHA8	
+	1.3	5.0	-2.0						

+			BULKTYP	BULKQ	BULKL				
+									

Field	Contents		Type	Default
MID	Unique material number		I > 0	Required
RHO	Density		R > 0.0	Required
NU	Poisson's ratio		0.0 ≤ R < 0.5	Required
Mui	Strain-energy density function constant		R	0.0
ALPHAi	Strain-energy density function constant		R	0.0
BULKTYP	Bulk-viscosity model		C	DYNA
	DYNA	Standard DYNA3D model		
BULKQ	Quadratic bulk-viscosity coefficient		R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient		R ≥ 0.0	0.0

Remarks

1. The fourth line with the bulk-viscosity data can be omitted.
2. The constitutive behavior of this material is defined as a total stress/total strain relationship. The nonlinear elastic material response is formulated by a strain-energy density function for large strain components using Ogden function.

   The strain-energy density function is formulated according to the Ogden model and is defined as

$$W = \sum_{i=1}^{i=1} \sum \frac{\mu_j}{\alpha_j}(\bar{\lambda}^{\alpha_j} - 1) + \frac{1}{2}K(J-1)^2$$

The over bar (-) indicates that the volumetric effects have been eliminated from the principal stretches.

3. This material can only be used with Lagrangian solid elements.
4. The behavior of this material is discussed in more detail in *Dytran Theory Manual*, Chapter 3: Materials.
5. When the elements with this material deform severely, it is possible that the job will be unstable. In this case, please use PARAM,HGTYPE,DYNA instead of default hourglass control.

SECTION

Cross Section

Defines a cross section of the model for force output.

Format and Example

1	2	3	4	5	6	7	8	9	10
SECTION	SID	GID	EID						
SECTION	101	5	8						

Field	Contents	Type	Default
SID	Unique cross-section number	I > 0	Required
GID	The number of a SET1 entry containing a list of grid points that define the cross section.	I > 0	Required
EID	The number of a SET1entry containing a list of elements that define the cross section.	I > 0	Required

Remarks

1. The cross sections for which output is required are referenced in a SET command in Case Control Section. The SET entry is referenced by the CSECS Case Control command.
2. The cross section is defined as a consecutive sequence of elements extending across the model. In addition, a consecutive sequence of grid points attached to one side of the elements must be defined. The GID field is required together with EID, the list of elements.
3. For compatibility with Dyna, the method of specifying three EIDs (i.e. one for one-dimensional elements, one for plate elements and one for hexahedral elements) is retained.
4. Cross sections cannot be defined for Eulerian models.

SET1

Set of Numbers

Defines a set of grid points, elements, etc., for use by other entries (e.g., WALL, SURFACE).

Format and Example

1	2	3	4	5	6	7	8	9	10
SET1	ID	N1	N2	N3	N4	N5	N6	N7	+CONT1
SET1	101	7	17	32	45	8	9	22	+CONT1
+CONT1	N8	N9	THRU	N10	BY	N11	-etc.-		
+CONT1	107	221	THRU	229	BY	3			

Field	Contents	Type	Default
ID	Number of the set	I > 0	Required
N1, N2, . . .	Numbers of the items in the set. If the word THRU appears between two numbers, all the numbers in the range are included in the list. BY indicates the increment within this THRU specification.	I > 0	Required

Remarks

1. Use as many continuation lines as necessary.
2. If the THRU specification is used, all the items in the sequence between the beginning and the end of the range do not have to exist. Items that do not exist are ignored. BY can be used as an increment to exclude grid points.
3. SET1 Bulk Data entries with the same number are merged into one set.

SETC

List of Names

Defines a list of names (character strings) for use by other entries.

Format and Example

1	2	3	4	5	6	7	8	9	10
SETC	ID	V1	V2	V3	V4	-etc.-			
SETC	100	HUB	RIM	HEAD	CHEST				

Field	Contents	Type	Default
ID	Unique SETC number	I > 0	Required
Vi	Character strings	C	Required

Remarks

1. Use as many continuations as required to define the complete list of names. A blank field terminates the list.
2. The SETC entry may be referred to from outside the Bulk Data Section.

SETTING

Application-Sensitive Defaults

Defines application-sensitive defaults for element formulation, element options, hourglass control and material behavior.

Format and Example

1	2	3	4	5	6	7	8	9	10
SETTING	SID	TYPE	PROP1	PID1	PROP2	PID2	PROP3	PID3	+
SETTING	100	CRASH	PCOMP	101	SHELL	102	SOLID	103	+
+	PROP4	PID4	PROP5	PID5	-etc.-				
+	SOLID	104	PCOMP						

Field	Contents		Type	Default
SID	Setting number		I > 0	Required
TYPE	Application type:		C	STANDARD See Remark 1.
	STANDARD	Standard defaults		
	CRASH	Defaults designed for crash simulations		
	SHEETMETAL	Defaults designed for sheet metal forming analysis		
	SPINNING	Defaults designed for fast rotating structures		
	FAST	Defaults for fast, but not necessarily the most accurate, solution		
	VERSION2	Defaults from Dytran V2.3		
PROPi	Property type		C	See Remark 2.
PIDi	Property number		I > 0	See Remark 2.

Remarks

1. The application-sensitive defaults are set according to the specification in the TYPE field. If no application type is specified, the setting is STANDARD. The default settings concern the element formulation, element formulation options, hourglass control, material-plasticity calculation method, and strain dependency of the thickness of shell elements. See *Dytran User's Guide*, Chapter 5: Application Sensitive Default Setting for more details on application-sensitive defaults.

2. If no property type and property number are supplied, the setting is done for all properties in the model. If the property type and the property number are defined, the setting applies to the elements that have the specified property. As such it is possible to define a global application setting and have a different setting for certain properties in the model.
3. See *Dytran User's Guide*, Chapter 5: Application Sensitive Default Setting for more details on application-sensitive defaults.

SHEETMAT

Sheet-Metal Material

Defines the properties of an anisotropic plastic material for Lagrangian shell elements. This material model uses Hill-48 yield criteria and allows the use of the Lankford parameters for definition of the anisotropy.

Format and Example

1	2	3	4	5	6	7	8	9	10
SHEETMAT	MID	RHO	EXX	EYY	EZZ	GXY	GYZ	GXZ	+
SHEETMAT	1	2.7E-6	72E6						+
+	NUXY	NUYZ	NUXZ	ELASTIC		XMAT	YMAT	ZMAT	+
+	0.33			ISO		1.0	0.0	0.0	+
+	a	b	c	n	k	m	TABLE		+
+	0.0	570E3	0.017	0.359	0.014	0.389			+
+	TYPEYLD	R0	R45	R90	TYPERM	CPTOL			+
+	PLANANI	0.73	0.51	0.69	2	1.0e-3			+
+	TYPEHRD								+
+	ISO								+
+	C1	C2	C3	C4	C5				+
+	0.244	-0.195	0.857	3.439	-11.92				+
+		D2	D3	D4	D5				
+		-0.417	-1.567	-4.849	-6.061				

Field	Contents	Type	Default
MID	Unique material number	I > 0	Required
RHO	Mass density	R > 0.0	Required
EXX, EYY, EZZ	Young's moduli in the X, Y and Z-direction (also defined as rolling, transverse and through-the-thickness directions, respectively)	R > 0.0	See Remark 2.
GXY	In-plane shear modulus	R > 0.0	See Remark 2.
GYZ, GXZ	Transverse shear moduli for shear in the YZ and XZ planes, respectively	R > 0.0	See Remark 2.

Field	Contents		Type	Default
NUXY, NUYZ, NUXZ	Poisson's ratios (coupled strain ratios in the XY, YZ, and XZ directions, respectively)		R ≥ 0.0	See Remark 2.
ELASTIC	Type of elasticity		C	ISO See Remark 3.
	ISO	ISOtropic material		
	PLANISO	PLANar ISOtropic material		
XMAT, YMAT, ZMAT	Vector indicating the rolling direction of the material		R	(0., 0., 0.) See Remark 4.
a	Power-law stress constant		R ≥ 0.0	Required See Remark 5.
b	Power-law hardening parameter		R ≥ 0.0	0.0
c	Power-law strain offset		R ≥ 0.0	0.0
n	Power-law, strain-hardening exponent		R ≥ 0.0	1.0
k	Power-law, strain-rate sensitivity constant		R ≥ 0.0	0.0 See Remark 6.
m	Power-law, strain-rate exponent		R ≥ 0.0	1.0
TABLE	Number of a TABLED1 entry giving the variation of yield stress (y-value) with effective plastic strain (x-value). If nonzero and positive, the curve will be used to define yield stress.		I>0	See Remark 5.
TYPEYLD	Type of yielding criterion.		C	ISO See Remark 7.
	ISO	ISOtropic yielding (von Mises)		
	NORMANI	NORMal ANIsotropic yielding		
	PLANANI	PLANar ANIsotropic yielding		
R0, R45, R90	Anisotropic yielding parameters (Lankford parameters) defined in 0, 45, and 90 degrees with respect to the rolling direction		R > 0.0	See Remark 8.
TYPERM	Type of return mapping algorithm		I	2. See Remark 9.
	1	Krieg's return mapping algorithm		
	2	Cutting-plane return mapping algorithm		

Field	Contents		Type	Default
CPTOL	Convergence tolerance for cutting plane algorithm		R≤1.0e-2	1.0e-6 See Remark 10.
TYPEHRD	Type of hardening rule.		C	ISO
	ISO	ISOtropic hardening		
	NORMANI	NORMal ANIsotropic hardening		
C1-C5	Engineering coefficients in limit function for e2 > 0.		R	C1 = 1.0 See Remark 11.
D2-D5	Engineering coefficients in limit function for e2 < 0.		R	0.0 See Remark 11.

Remarks

1. SHEETMAT materials may only be referenced by PSHELL and PSHELL1 entries.
2. The necessary number or combination of elasticity constants depends on the field ELASTIC. If ELASTIC = ISO then only EXX and NUXY (or GXY) must be defined. For ELASTIC = PLANISO, only EXX (or EYY), EZZ, NUXY (or GXY), NUXZ (or NUYZ), and GXZ (or GYZ) must be defined.
3. The field ELASTIC provides you with an input check on the consistency of the elasticity constants. Planar isotropic material is equivalent to transversely isotropic material, which means that the through-the-thickness (elastic) properties may differ from the in-plane isotropic (elastic) properties.
4. Due to anisotropic behavior, the rolling direction must be specified.
   - XHAT, YHAT, ZHAT = (0.,0.,0.): Base axis is node 1 to node 2 in the element connectivity table + THETA on the CQUAD4 and CTRIA3 entries.
   - XHAT, YHAT, ZHAT: ≠ (0.,0.,0.): The projection of the vector (XMAT, YMAT, ZMAT)on the surface of each element is used to determine the angle between the element and the material coordinate system. This angle can be overwritten using the THETA field on the CQUAD4 and CTRIA3 entries.

   Both the constitutive law and output of variables are applied with respect to this material coordinate system (see Remark 12.).

   For example:

Case	CTRIA3/CQUAD4 THETA	XMAT/YMAT/ZMAT	Rolling Direction
1	THETA	(X,Y,Z)	THETA in CTRIA3/CQUAD4
2	0.0	(X,Y,Z)	(X,Y,Z) in XMAT/YMAT/ZMAT

Case	CTRIA3/CQUAD4 THETA	XMAT/YMAT/ZMAT	Rolling Direction
3	Blank	(X,Y,Z)	(X,Y,Z) in XMAT/YMAT/ZMAT
4	THETA	(0,0,0)	THETA in CTRIA3/CQUAD4
5	0.0	(0,0,0)	0.0 in CTRIA3/CQUAD4
6	Blank	(0,0,0)	0.0 in CTRIA3/CQUAD4

5. For a description of the anisotropic-plastic model including hardening, see *Dytran Theory Manual*, Chapter 3: Materials. The hardening of material can be introduced by providing the variation of yield stress with respect to the effective plastic strain. This relationship can be given using either a table or the power-law function and not both. The power-law stress constant, `a`, is not necessarily the initial yield stress: the value of `a` is allowed to be equal to zero if the value of the hardening parameter, `b`, and the strain offset, `c`, are unequal to zero.
6. Strain-rate dependence is not accounted for by default.
7. The field `TYPEYLD` provides you with an input check on the consistency of the anisotropic parameters. Normal anisotropic material is equivalent to transversely anisotropic or planar isotropic material which means that the through-the-thickness yielding properties may differ from the in-plane, isotropic, yielding properties. Planar anisotropic material is characterized by three orthogonal axes of anisotropy (in rolling, transverse and through-the-thickness direction), about which the yielding properties have twofold symmetry.
8. The necessary number of anisotropic-yielding parameters depends on the field `TYPEYLD`. For `TYPEYLD = ISO`, all fields for `R0`, `R45`, and `R90` can be left blank because the default corresponds to von Mises yielding (R0 = R45 = R90 = 1.0). For `TYPEYLD = NORMANI`, only R0 must be defined while the other two fields can be left blank due to their equality. The input of all three anisotropic parameters is needed for `TYPEYLD = PLANANI`.
9. In Dytran 2021, the cutting-plane return mapping algorithm is added in SHEETMAT material along with the existing Krieg's return mapping algorithm. The new algorithm provides more accurate results, however at a higher computational cost. The default return mapping algorithm is set to cutting-plane algorithm, so, to reproduce the same results as before Dytran 2021, please set `TYPERM=1`.
10. Krieg's return mapping algorithm does not require `CPTOL`.
11. `C1` through `C5` and `D2` through `D5` do not affect the material behavior but are used to fit the lower bound of experimental results for diffuse and localized necking represented by two polynomial lines:

$$FLD(e_2) = C_1 + C_2 e_2 + C_3 e_2^2 + C_4 e_2^3 + C_5 e_2^4 \text{ for } e_2 > 0.0$$

$$FLD(e_2) = C_1 + D_2 e_2 + D_3 e_2^2 + D_4 e_2^3 + D_5 e_2^4 \text{ for } e_2 < 0.0$$

12. The output of variables related to SHEETMAT is defined with respect to the material coordinate system (see Remark 4.). There are a number of specific output variables useful for this material:

Element Variables	
Q1, Q2	Direction cosines/sines between the element coordinate system and the material coordinate system
Sublayer Variables	
TXX	Stress - XX component
TYY	Stress - YY component
TXY	Stress - XY component
TYZ	Stress - YZ component
TZX	Stress - ZX component
EFFST	Effective Stress
EFFPL	Effective Plastic Strain
YLDRAD	Radius of Yield Surface
EPSXX	Strain - XX component
EPSYY	Strain - YY component
EPSXY	Strain - XY component
EPSZZ	Strain - ZZ component
EPZZ	Plastic Strain – ZZ component
EPSMX	Strain - Major Principal Strain
EPSMN	Strain - Minor Principal Strain
FLP	Forming-Limit Parameter

SHREL

Elastic Shear Model

Defines an elastic shear model with a constant shear modulus.

Format and Example

1	2	3	4	5	6	7	8	9	10
SHREL	SID	G							
SHREL	250	80.E6							

Field	Contents	Type	Default
SID	Unique shear model number referenced from a DMAT entry	I > 0	Required
G	Shear-modulus value	R	0.0

Remark

Shear model numbers must be unique.

SHREX

User-specified Shear Model

Specifies that a user subroutine is being used to define the shear modulus.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE mymat SCA.MDSolver.Obj.Uds.Dytran.Flow
```

Format and Example

1	2	3	4	5	6	7	8	9	10
SHREX	SID	GROUP							
SHREX	20	mymat							

Field	Contents	Type	Default
SID	Unique shear model number referenced from a DMATentry	I > 0	Required.
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. See Chapter 7: User Defined Services for a description of how to use user-written subroutines.
2. This shear model is applicable only for Lagrangian solid elements and Eulerian elements with shear strength.

SHRLVE — Isotropic Linear Viscoelastic Shear Model

Defines an isotropic linear viscoelastic shear model where the mechanical analog is a spring, a dashpot, and a Maxwell element connected in parallel.

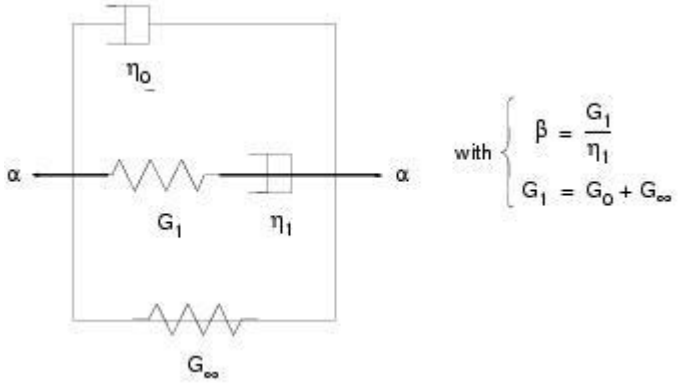
Format and Example

1	2	3	4	5	6	7	8	9	10
SHRLVE	SID	G_0	G_∞	β	η_0				
SHRLVE	250	8.E7	1.E7	0.1					

Field	Contents	Type	Default
SID	Unique shear model number referenced from a DMAT entry	I > 0	Required
G_0	Short-time shear modulus	R	0.0
G_∞	Long-time shear modulus	R	0.0
β	Decay constant	R	0.0
η_0	Shear viscosity constant	R	0.0

Remarks

1. Shear model numbers must be unique.
2. The spring-damper analog of this model is



3. The deviatoric stress is given by

$$= 2G_{\infty}\varepsilon'_{ij}(t) + 2\int_{0} G(t-\tau) + \frac{\partial\varepsilon'_{ij}(\tau)}{\partial\tau}d\tau + 2\eta_0$$

with the relaxation function

$$G(t-\tau) = (G_0 - G_{\infty})e^{-\beta(t-\tau)}$$

The above equation for the deviatoric stress is the integral form of the differential equation

$$\dot{\sigma}'_{ij} + \beta\sigma'_{ij} = 2\eta_0\ddot{\varepsilon} + (2\eta_0\beta + 2G_0)\dot{\varepsilon} + 2G_{\infty}\beta\varepsilon$$

A special case is $\eta_0 = G_{\infty} = 0$, for which is often written

$$\dot{\varepsilon} = \dot{\varepsilon}_{elastic} + \varepsilon_{viscous} = \frac{\dot{\sigma}'_{ij}}{2G_0} + \frac{\beta}{2G_0}\sigma'_{ij}$$

This shear model is further described in *Dytran Theory Manual*, Chapter 4: Models, Shear Models.

4. A yield model cannot be used in combination with this shear model.
5. The element formulation for this material is in a corotational frame. The default CORDROT definition is G1 = 1, G2 = 5, G3 = 2. (See also DMATand CORDROT entries.)

SHRPOL

Polynomial Shear Model

Defines an elastic shear model with a polynomial shear modulus.

Format and Example

1	2	3	4	5	6	7	8	9	10
SHRPOL	SID	G0	G1	G2	G3				
SHRPOL	250	180.e6							

Field	Contents	Type	Default
SID	Unique shear model number referenced from a DMATentry	I > 0	Required
G_0	Coefficient G_0	R	0.0
G_1	Coefficient G_1	R	0.0
G_2	Coefficient G_2	R	0.0
G_3	Coefficient G_3	R	0.0

Remarks

1. Shear model numbers must be unique.
2. The shear modulus is computed from

$$G = G_0 + G_1\gamma + G_2\gamma^2 + G_3\gamma^3$$

where γ = effective plastic shear strain

and G_0, G_1, G_2, and G_3 are constants

SPC

Single-Point Constraint

Defines sets of single-point constraints.

Format and Example

1	2	3	4	5	6	7	8	9	10
SPC	SID	G	C		G	C			
SPC	2	32	436		5	1			

Field	Contents	Type	Default
SID	Number of single-point constraint sets	I > 0	Required
G	Grid-point number	I > 0	Required
C	Component number of global coordinate (any unique combination of the digits 1 through 6 with no embedded blanks). Combinations are allowed; e.g., 23, 156.	I > 0	Required

Remarks

1. SPC degrees of freedom may also be specified as permanent constraints on the GRID entry.
2. Continuation lines are not allowed.
3. Select single-point constraints in the Case Control Section (SPC = SID) to be used by Dytran.
4. A single-point constraint is treated as a zero-velocity boundary condition. For this reason, make SPCs consistent with other velocity boundary conditions and velocity initial conditions.

SPC1

Single-Point Constraint

Defines a single-point constraint for a set of grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
SPC1	SID	C	G1	G2	G3	G4	G5	G6	+
SPC1	3	2	1	3	10	9	6	5	+
+	G7	G8	THRU	G10	BY	G11	-etc.-		
+	2	8	THRU	24	BY	3			

Field	Contents	Type	Default
SID	Number of a single-point constraint.	I > 0	Required
C	Component number of global coordinate (any unique combination of the digits 1 through 6 with no embedded blanks). Combinations are allowed; e.g., 12, 456.	I > 0	Required
Gi	Grid-point numbers. THRU indicates a range of grid points. BY is the increment within this range.	I > 0	Required

Remarks

1. As many continuation lines as desired may appear.
2. SPC degrees of freedom may be redundantly specified as permanent constraints on the GRIDentry.
3. If the THRU specification is used, grid points in the sequence between the beginning and the end of the range are not required. Grid points that do not exist are ignored. BY can be used to exclude grid points within this range.
4. Single-point constraints must be selected in the Case Control Section (SPC = SID) if they are to be used by Dytran.
5. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.
6. A single-point constraint is treated as a zero-velocity boundary condition. For this reason, make SPCs consistent with other velocity boundary conditions and velocity initial conditions.

SPC2

Single-Point Constraint

Rotational boundary constraint on grid points.

Format and Example

1	2	3	4	5	6	7	8	9	10
SPC2	SID	G		TYPE1	VALUE1	NX	NY	NZ	+
SPC2	12	302		TABLE	410	1.0	0.0	0.0	+
+	TYPE2	VALUE2							+
+	CONSTANT	0.							+
+	G1	G2	THRU	G3	BY	G5	-etc-		
+	10	13	THRU	56	BY	4			

Field	Contents		Type	Default
SID	Number of a single-point constraint		I > 0	Required
G	Grid-point number of a point on the rotation axis		I > 0	Required
TYPE1	Defines the type of rotational constraint.		C	Required
	CONSTANT	The rotational (angular) velocity is constant at VALUE1 times the length of the rotation vector.		
	TABLE	The rotational (angular) velocity varies with time as the interpolated value in TABLED1 with number VALUE1, times the rotation vector magnitude.		
VALUE1	Value depending on TYPE1		R or I > 0	Required
NX, NY, NZ	Rotation vector		R	Required
TYPE2	Defines the type of radial constraint:		C	Required
	CONSTANT	The radial velocity is constant at VALUE2 where VALUE2 must be zero.		
	FREE	The radial velocity is free and determined by the forces in the direction of the radius. The VALUE2 entry is ignored.		
VALUE2	Value depending on TYPE2		R	Required
Gi	Grid-point numbers. THRU indicates a range of grid points. BY is the increment to be used within this range. (G2 < G3)		I > 0	Required

Remarks

1. The angular velocity is specified in radians per unit time.
2. The SPC2 entry is valid for both Lagrangian as Eulerian grid points.
3. If the TYPE2 field is set to FREE, the referenced grid points move in a radial direction according to the acceleration caused by forces in the radial direction.
4. You can use as many continuation lines as required.
5. If the THRU specification is used, grid points in the sequence between the beginning and the end of the range are not required to exist. Grid points that do not exist are ignored.
6. Select the rotational constraints in the Case Control Section (SPC = SID) if they are to be used by Dytran.
7. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.
8. Both Lagrangian and Eulerian grid points can have a rotational constraint. In the case of Eulerian grid points, this results in a moving Eulerian mesh.
9. For six degree of freedom grid points, the angular velocities are also constrained consistent with the defined velocity field.
10. The velocity in axial direction is constrained to zero.
11. The X, Y, and Z location of node G is used to determine the axis of rotation and this remains constant in time. Subsequently, if the node G moves during the simulation, the axis of rotation will not move with it, but remain stationary.

SPC3

Single-Point Constraint

Defines a single-point constraint in a local coordinate system or a cascade of two local coordinate systems.

Format and Example

1	2	3	4	5	6	7	8	9	10
SPC3	SID	CID1	C1	CID2	C2				+
SPC3	1	5	12						+
+	G1	G2	THRU	G3	BY	G4	-etc.-		
+	5	6	THRU	18	BY	3			

Field	Contents	Type	Default
SID	Number of a single-point constraint	I > 0	Required
CID1	Number of the primary coordinate system	I > 0	See Remark 7.
C1	Constraint with respect to CID1	I > 0	See Remark 7.
CID2	Number of the secondary coordinate system	I > 0	See Remarks 7. and 11.
C2	Constraint motion of primary coordinate system CID1 with respect to CID2	I > 0	See Remark 7.
Gi	Grid-point numbers. THRU indicates a range of grid points. BY is the increment to be used within this range.	I > 0	Required

Remarks

1. CID1 references the primary system. In this system the grid point constraints are applied. The CID2 system defines a secondary system that constrains the motion of the primary system and the grid points defined on the entry.
2. The SPC3 entry is valid for both Lagrangian as Eulerian grid points.
3. As many continuation lines as desired may appear.
4. If the THRU specification is used, grid points in the sequence between the beginning and the end of the range are not required. Grid points that do not exist are ignored. (See Remark 3. of SPC1.)
5. Select the single-point constraint in the Case Control Section (SPC = SID) if it is to be used by Dytran.
6. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.
7. If CID1 or CID2 is blank, the basic system is used. If C1 is blank, no constraints are applied in the primary system. If C2 is blank, no constraints are applied in the primary system with respect to the secondary system.
8. If CID1, CID2, and C2 are left blank, the constraint acts as defined by an SPC1 entry.

9. If a component references an angular velocity, the units are radians per unit time.
10. A single-point constraint is treated as a zero velocity boundary condition. For this reason, make SPCs consistent with other velocity boundary conditions and velocity initial conditions.
11. CID2 = 0 is not allowed. Instead define a new local coordinate system at (0., 0., 0.) and refer to this coordinate system on CID2.

SPHERE

Defines the Shape of a Sphere

Spherical shape used in the initial condition definition on the TICEUL entry.

Format and Example

1	2	3	4	5	6	7	8	9	10
SPHERE	VID		X	Y	Z	RADIUS			
SPHERE	100		1.	1.	1.	.5			

Field	Contents	Type	Default
VID	Number of the sphere	I > 0	Required
X, Y, Z	Coordinates of the center of the sphere	R	0.0
RADIUS	Radius of the sphere	R > 0.0	Required

SUBSURF

Multifaceted Subsurface

Defines a multifaceted subsurface for contact and coupling interfaces.

Format and Example

1	2	3	4	5	6	7	8	9	10
SUBSURF	SSID	SID	TYPE1	SID1	TYPE2	SID2	TYPE3	SID3	+
SUBSURF	100	100	ELEM	10	PROP	20	SEG	30	+
+	TYPE4	SID4	-etc.-						
+	MAT	100							

Field	Contents		Type	Default
SSID	Unique subsurface number		I > 0	Required
SID	Number of a SURFACEentry of which these segments are a subsurface		I > 0	Required
TYPEi	The type of entity used to define the subsurface.		C	Required
	SEG	A set of segments defined using CSEGCFACE, or CFACE1 entries. SIDi is the set number of the segments.		
	ELEM	A set of segments attached to shell and/or membrane elements and selected by the element number. SIDi is the number of a SET1 entry containing a list of the element numbers to be used.		
	PROP	A set of segments attached to shell and/or membrane elements and selected by the property number. SIDi is the number of a SET1 entry containing a list of the property numbers to be used.		
	MAT	A set of segments attached to shell and/or membrane elements and selected by material number. SIDi is the number of a SET1 entry containing a list of the material numbers to be used.		
SIDi	The number of a set of CSEG CFACE, or CFACE1entries or the number of a SET1 entry, depending on the value of TYPEi.		I > 0	Required

Remarks

1. You can use as many continuation lines as necessary to define all of the segments in the surface.
2. CSEGs are defined indirectly using CQUAD4 and/or CTRIA3elements with a 9999. thickness. CFACE1 are entries defined indirectly using PLOAD4 entries with a 9999. pressure. This allows CSEG and CFACE1 entries to be easily defined using standard preprocessors that can generate CQUAD4, CTRIA3 and PLOAD4 entries.
3. The subsurface SSID can be referenced from the following entries:

SURFACE	To define a surface that has the same segments as this subsurface.
CONTINI	To define the initial contact between Lagrangian surfaces. The surface SID must then be used in a CONTACT entry.
COUPOR	To define the porosity of a coupling surface. The surface SID must then be used in a COUPLE entry.
COUOPT	To define the options used in a coupling surface. The surface SID must then be used in a COUPLE entry.

SURFACE

Multifaceted Surface

Defines a multifaceted surface for contact and coupling interfaces as well as rigid-surface geometry.

Format and Example

1	2	3	4	5	6	7	8	9	10
SURFACE	SID		TYPE1	SID1	TYPE2	SID2	TYPE3	SID3	+
SURFACE	100		ELEM	10	PROP	20	SEG	30	+
+	TYPE4	SID4	TYPE5	SID5	-etc.-				
+	MAT	100	SUB	200					

Field	Contents		Type	Default
SID	Unique surface number		I > 0	Required
TYPEi	The type of entity used to define the surface:		C	Required
	SEG	A set of segments defined using CSEG, CFACE, or CFACE1 entries. SIDi is the set number of the segments.		
	ELEM	A set of segments attached to shell and/or membrane elements and selected by element number. SIDi is the number of a SET1 entry containing a list of the element numbers to be used.		
	PROP	A set of segments attached to shell and/or membrane elements and selected by property number. SIDi is the number of a SET1 entry containing a list of the property numbers to be used.		
	MAT	A set of segments attached to shell and/or membrane elements and selected by material number. SIDi is the number of a SET1 entry containing a list of the material numbers to be used.		
	SUB	A set of segments defined by a SUBSURF entry. SIDi is the number of the SUBSURF entry.		
SIDi	The number of a set of CSEG, CFACE, or CFACE1 entries, the number of a SET1 entry or the number of a SUBSURFentry depending on the value of TYPEi.		I > 0	Required

Remarks

1. You can use as many continuation lines as necessary to define all of the segments in the surface.
2. CSEGs are defined indirectly using CQUAD4 and/or CTRIA3elements with a 9999. thickness. CFACE1are entries defined indirectly using PLOAD4entries with a 9999. pressure. This allows CSEG and CFACE1 entries to be easily defined using standard preprocessors that can generate CQUAD4 CTRIA3, and PLOAD4 entries.

TABFILE

Text File Defined Function

Specifies that a text file is used to define a tabular function.

Format and Example

1	2	3	4	5	6	7	8	9	10
TABFILE	ID	NAME							
TABFILE	2	MYFILE							

Field	Contents	Type	Default
ID	Unique table number.	I > 0	Required
NAME	File name (no longer than 80 characters).	C	None

Remark

The text file has to consist of a number of data lines and may include comment lines. Each comment line has to start with $. Every data line has to consist of two numbers. These will be interpreted as XVALUE and the YVALUE of a TABLED1. Examples 3.4 and 3.5 illustrate the use. The table can be checked in the Out file. Blank lines are not supported. The first line of a file can start with XYDATA. It will be ignored.

Example

```
TABFILE,200,rho.xyd
```

```
 Contents rho.xyd:
$ density as function of radius
 0.00125 0.50041467
 0.0037499995 0.52369827
 0.0062499996 0.52935588
```

TABLED1

Table

Defines a tabular function.

Format and Example

1	2	3	4	5	6	7	8	9	10
TABLED1	ID								+
TABLED1	32								+
+	X1	Y1	X2	Y2	X3	Y3	X4	Y4	+
+	-3.0	6.9	2.0	5.6	3.0	5.6	XSMALL	ENDVAL	+
+	X5	Y5	X6	Y6	X7	Y7	X8		
+	XLARGE	EXTRAP	XOFFSET	.05	YOFFSET	.04	ENDT		

Field	Contents	Type	Default
ID	Unique table number	I > 0	Required
Xi, Yi	Tabular entries. Special entries for Xi, Yi are given in Remark 6.	R or C	0.0

Remarks

1. The values of Xi must be in ascending or descending order but not both.
2. At least two entries must be present.
3. The end of the table is marked by the characters ENDT in the field following the last table entry or by a blank field.
4. The table is used according to

   $$y = f(x)$$

   where x is input to the table and y is output. Linear interpolation is used within the table to determine y. Outside the table, the last entry for y is taken.

5. Instead of a numerical value for a y entry, the keyword FREE can be entered. The value of FREE in the table can be used together with constraints and loading to switch these on and off. FREE means that the constraint or loading is not active during the time interval for which the FREE entry is defined.

6. Special entries can be given for `Xi`,`Yi` to specify:

   Extrapolation outside x-range or not.

   Offset for the x- and y-axis.

   Scale factor for the x- and y-axis:

Xi	Yi	Meaning
XSMALL	`ENDAVL` or	If x if outside the table from the smallest value, the first entry for y is taken.
	`EXTRAP`	If x is outside the table from the smallest value, y is extrapolated.
XLARGE	`ENDAVL` or	If x is outside the table from the largest value, the last entry for y is taken.
	`EXTRAP`	If x is outside the table from the largest value, y is extrapolated.
XOFFSET	value	x-axis of table is offset by the specified value.
YOFFSET	value	y-axis of table is offset by the specified value.
XSCALE	value	x-axis of table is multiplied by the specified value.
YSCALE	value	y-axis of table is multiplied by the specified value.

TABLEEX

User-defined Function

Specifies that a user routine is being used to define an arbitrary function.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE myfunc SCA.MDSolver.Obj.Uds.Dytran.Loads
```

Format and Example

1	2	3	4	5	6	7	8	9	10
TABLEEX	ID	NAME	GROUP						
TABLEEX	2	MYTABLE	myfunc						

Field	Contents	Type	Default
ID	Unique table number	I > 0	Required
NAME	Name of the function (no longer than 16 characters)	C	None
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. See Chapter 7: User Defined Services in this manual for a description of how to use user-written subroutines.
2. Since tables and user-defined functions belong to the same group, the table numbers must be unique.

TIC

Transient Initial Velocities of Grid Points

Defines the initial velocities of Lagrangian grid points at the beginning of the analysis.

Format and Example

1	2	3	4	5	6	7	8	9	10
TIC	SID	G	C		V0				
TIC	1	3	2		-13.3				

Field	Contents	Type	Default
SID	Set number	I > 0	Required
G	Grid-point number to be initialized	I > 0	Required
C	Component number (a digit 1 through 6)	$1 \le I \le 6$	Required
V0	Initial velocity value	R	0.0

Remarks

1. Initial conditions for grid points that are not specified on TICn or TICGP entries are assumed to be zero.
2. Initial conditions to be used by Dytran must be selected in the Case Control Section (TIC = SID).
3. Only Lagrangian grid points can have initial conditions specified by the TIC Bulk Data entry.
4. Only one TIC entry per grid point is allowed. When more than one velocity component needs to be initialized, TICGP offers a more general way of initializing grid-point velocities.

TIC1

Transient Initial Velocities of Grid Points

Defines the initial velocities of Lagrangian grid points at the beginning of the analysis.

Format and Example

1	2	3	4	5	6	7	8	9	10
TIC1	SID	C		V0	G1	G2	G3	G4	+
TIC1	3	2		3	10	9	6	5	+
+	G5	G6	THRU	G7	BY	G8	-etc.-		
+	2	8	THRU	17	BY	3			

Field	Contents	Type	Default
SID	Set number	I > 0	Required
C	Component number (a digit 1 through 6)	$1 \le I \le 6$	Required
V0	Initial velocity value	R	0.0
G1, G2, ...	Grid-point numbers to be initialized. If the word THRU appears between two numbers, all the grid points in the range are initialized. BY indicates an increment within this range.	I > 0	Required

Remarks

1. Initial conditions for grid points that are not specified on TICn or TICGP entries are assumed to be zero.
2. Only one TIC1 entry per grid point is allowed. When more than one velocity component needs to be initialized, TICGP offers a more general way of initializing grid-point velocities.
3. If the THRU specification is used, all grid points in the sequence between the beginning and the end of the range do not have to exist. Grid points that do not exist are ignored. The first grid point in the THRU specification must be a valid grid point. BY enables grid points to be ignored in this range.
4. None of the fields in the list of grid points can be blank or zero since this marks the end of the list. The initial conditions to be used by Dytran must be selected in the Case Control Section (TIC = SID).
5. Only Lagrangian grid points can have initial conditions specified by the TIC1 Bulk Data entry.

TIC2 Transient Initial Velocities of Grid Points

Defines the initial velocities of grid points consistent with a rotational field.

Format and Example

1	2	3	4	5	6	7	8	9	10
TIC2	SID	G		SCALE	NX	NY	NZ		+
TIC2	3	1		10.	0.1	0.2	0.3		+
+	G1	G2	THRU	G3	BY	G4	-etc.-		
+	1	2	THRU	10000	BY	23			

Field	Contents	Type	Default
SID	Number of a set of loads	I > 0	Required
G	Number of a grid point on the axis of rotation	I > 0	Required
SCALE	Scale factor for the rotational velocity	R	1.0
NX, NY, NZ	Components of the rotation direction vector. The vector acts at point G.	R	Required
G1, G2, ...	Grid points to be initialized. THRU indicates a range of grid points. BY is the increment to be used within this range.	I > 0	Required

Remarks

1. The rotational velocity w is calculated as:

   $$w = SCALE*N$$

   where $SCALE$ is the scale factor and $\underline{N}$ is the vector defined by NX, NY, NZ.
2. Any number of TIC2 entries can be used.
3. The rotational velocity is defined in radians per unit time.
4. For six degree of freedom grid points, the angular velocities are also initialized.
5. Initial conditions for grid points that are not specified on TIC or TICGP entries are assumed to be zero.
6. If the THRU specification is used, all grid points in the sequence between the beginning and the end of the range do not have to exist. Grid points that do not exist are ignored. The first grid point in the THRU specification must be a valid grid point. BY enables grid points to be ignored in this range.
7. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.
8. The initial conditions to be used by Dytran must be selected in the Case Control Section (TIC = SID).

TIC3 General Form of Transient Initial Velocities of Grid Points

Allows for the definition of a velocity field of grid points consisting of a rotation and a translation specification.

Format and Example

1	2	3	4	5	6	7	8	9	10
TIC3	SID	G		SCALE					+
TIC3	7	5		10.					+
+	XVEL	YVEL	ZVEL	XROT	YROT	ZROT			+
+	100.0			5.0		-7.5			+
+	G1	G2	THRU	G3	BY	G4	-etc.-		
+	1	2	THRU	1000	BY	23			

Field	Contents	Type	Default
SID	Number of a set of loads	I > 0	Required
G	Number of a grid point at the center of rotation	I > 0	Required
SCALE	Apply to both translational and rotational velocity components.	R	1.0
XVEL, YVEL, ZVEL	Initial translational velocity components.	R	0.0
XROT, YROT, ZROT	Initial rotational velocity components.	R	0.0
G1, G2, ...	Grid points to be initialized. THRU indicates a range of grid points. BY is the increment to be used within this range.	I > 0	Required

Remarks

1. Any number of TIC3 entries can be used.
2. The rotational velocity components are defined in radians per unit time.
3. For six degree of freedom grid points, the angular velocity components are also initialized.
4. Initial conditions for grid points that are not specified on TIC or TICGP entries are assumed to be zero.
5. If the THRU specification is used, the grid points in the range definition are not required to exist. Grid points that do not exist are ignored. The first grid point in the THRU specification must be a valid grid point. The BY option enables grid points to be ignored in this range.None of the fields in the list of grid points can be blank or zero, since this indicates the end of the list.
6. The initial conditions to be used by Dytran must be selected in the Case Control Section (TIC = SID).

TICEEX User-defined Transient Initial Conditions of Elements

Defines the initial values of element variables at the beginning of the analysis by a user-written subroutine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE minit SCA.MDSolver.Obj.Uds.Dytran.InitOut
```

Format and Example

1	2	3	4	5	6	7	8	9	10
TICEEX	SID	SETID	NAME	GROUP					
TICEEX	2	20	INEL1	minit					

Field	Contents	Type	Default
SID	Set number	I > 0	Required
SETID	Number of a SET1 entry defining the elements to be initialized	I > 0	Required
NAME	Initial condition name passed to the user-written subroutin	C	None
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. See Chapter 7: User Defined Services in this manual for a description of how to use user-written subroutines.
2. Initial conditions to be used by Dytran must be selected in the Case Control Section (TIC = SID).

TICEL

Transient Initial Conditions of Elements

Defines the initial values of element variables at the beginning of the analysis.

Format and Example

1	2	3	4	5	6	7	8	9	10
TICEL	SID	SETID	NAME1	VALUE1	NAME2	VALUE2	-etc.-		
TICEL	3	40	DENSITY	100.	SIE	1.E5			

Field	Contents	Type	Default
SID	Set number	I > 0	Required
SETID	Number of a SET1 entry defining the elements to be initialized.	I > 0	Required
NAMEi	Element variable to be initialized. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Running Dytran Using the dytran Command.	C	Required
VALUEi	Value of the variable	R	Required

Remarks

1. Initial conditions for elements that are not specified on TICEL entries are assumed to be zero except density, which is set to the reference density.
2. Initial conditions to be used by Dytran must be selected in the Case Control Section (TIC = SID).
3. As many continuation lines as required can be used to specify all the variables being initialized. A blank field terminates the list.
4. Element variables for Eulerian elements can be initialized with a TICEL or a TICEULentry. The TICEL entry initializes a set of elements, while theTICEUL entry initializes either a set of elements or geometrical regions (sphere, cylinder,...). When a Euler element is part of both a TICEL and a TICEUL entry, the TICEL entry takes precedence, and overrules the TICEUL initialization for the element.

TICEUL Transient Initial Conditions of Eulerian Regions

Defines the initial value sets for Eulerian regions. The Eulerian regions are defined by geometric shapes.

Format and Example

1	2	3	4	5	6	7	8	9	10
TICEUL	SID								+
TICEUL	300								+
+	TYPE1	VID1	MID1	TSID1	LEVEL1				+
+	SPHERE	400	100	3	4.0				+
+	TYPE2	VID2	MID2	TSID2	LEVEL2				+
+	ELEM	500	200	4	2.1				+
+	TYPE3	VID3	MID3	TSID3	LEVEL3				+
+	CYLINDER	300	300	5	1.0				+
+	TYPE4	VID3	MID3	TSID3	LEVEL3				+
+	BOX	600	400	1	3.0				+
+	TYPEi	VIDi	MIDi	TSIDi	LEVELi	-etc-			
+	SURF	700	500	2	5.0				

Field	Contents		Type	Default
SID	Unique TICEUL number referenced from a PEULER1 entry		I > 0	Required
TYPEi	The type of Eulerian region:		C	Required
	SURF	Region inside or outside a multifaceted surface		
	SPHERE	Region inside a sphere		
	CYLINDER	Region inside a cylinder		
	BOX	Region inside a box		
	ELEM	Region defined by element list		
VIDi	Number of a geometric entity, a SET1number, or number of a MATINIentry		I > 0	Required
MIDi	Number of a DMATentry defining this material		I > 0	Required

Field	Contents	Type	Default
TSIDi	Unique TICVAL number containing a list of initial values for this material	I > 0	Required
LEVELi	Level indicator for this material and initial values.	R	0.0

Remarks

1. When the material number is left blank or zero, the Eulerian elements inside the region will be void.
2. All level indicators LEVELi must have different values. The level indicator can be negative.
3. See also the MICRO parameter for the accuracy of the initial value generation.
4. See *Dytran User's Guide*, Chapter 3: Constraints and Loading, Eulerian Loading and Constraints for instructions on how to use the geometric shapes on the TICEUL entry for arbitrary initial value generation in Eulerian regions.
5. Element variables for Eulerian elements can be initialized with a TICELor a TICEUL entry. The TICEL entry initializes a set of elements, while the TICEUL entry initializes either a set of elements or geometrical regions (sphere, cylinder, box, ...). When an Euler element is part of both a TICEL and a TICEUL entry, theTICEL entry takes precedence and overrules the TICEUL initialization for the element.
6. For the Euler solvers, you can, in addition to the "normal" element variables that the solver has defined, also define an initial radial velocity field. You have to enter the location of the center from where the radial emerges, the velocity to be applied to the element center and the decay coefficient for the velocity field. The center is defined by the keywords X-CENTER, Y-CENTER, Z-CENTER, the radial velocity by R-VEL and the decay coefficient by DECAY. You have to input these keywords in the above order, and have every keyword followed by its value. For more information, please refer to the *Dytran User's Guide*, Chapter 3: Constraints and Loading, Eulerian Loading and Constraints and the TICVAL or TICEL information in this manual.

TICGEX User-defined Transient Initial Conditions of Grid Points

Defines the initial values of grid-point variables at the beginning of the analysis by a user-written subroutine.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE minit SCA.MDSolver.Obj.Uds.Dytran.InitOut
```

Format and Example

1	2	3	4	5	6	7	8	9	10
TICGEX	SID	SETID	NAME	GROUP					
TICGEX	4	40	INGP3	minit					

Field	Contents	Type	Default
SID	Set number	I > 0	Required
SETID	Number of a SET1 entry defining the grid points to be initialized	I > 0	Required
NAME	Initial condition name passed to the user-written subroutine	C	None
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. See Chapter 7: User Defined Services in this manual for a description of how to use user-written subroutines
2. Initial conditions must be selected in the Case Control Section (TIC = SID) to be used by Dytran.

TICGP — Transient Initial Conditions for Grid Points

Defines the initial conditions of grid points at the beginning of the analysis.

Format and Example

1	2	3	4	5	6	7	8	9	10
TICGP	SID	SETID	NAME1	VALUE1	NAME2	VALUE2	-etc.-		
TICGP	3	30	PMASS	100.0	YVEL	30.0			

Field	Contents	Type	Default
SID	Transient initial condition set number	I > 0	Required
SETID	Number of a SET1 entry listing the grid points to be initialized	I > 0	Required
NAMEi	Grid-point variable to be initialized (see *Dytran User's Guide*, Chapter 9: Running the Analysis, Result Types) or CID1, CID2 (see Remark 4.)	C	Required
VALUEi	Value of the grid point variable, or number of coordinate system CID1, CID2 (see Remark 4.)	I or R	Required

Remarks

1. Initial conditions for grid-point components that are not specified on TIC or TIGGP entries are assumed to be zero.
2. Select initial conditions to be used by Dytran in the Case Control Section (TIC = SID).
3. Use as many continuation lines as required to specify all the variables being initialized. A blank field terminates the list.
4. The NAMEi on the TICGP entry can also be CID1 or CID2. In that case, VALUEi denotes the number of a defined coordinate system. Velocities are initialized according to the type of defined coordinate system. If coordinate systems are used, the velocity components must follow the CID definition immediately. All other variables must be defined before the first CID definition. Only for Lagrangian grid points the velocity can be defined in a local coordinate system.

   For example:

   ```
   TICGP, 1, 1,PMASS,10.,CID1,1,YVEL,10.
   ```
5. All velocity components defined and preceding a coordinate system reference are overruled by the definition following the coordinate system reference.

TICVAL

Transient Initial Condition Set

Defines the initial values of an Eulerian geometric region.

Format and Example

1	2	3	4	5	6	7	8	9	10
TICVAL	TSID	METHOD	NAME1	VALUE1	NAME2	VALUE2	NAME3	VALUE3	+
TICVAL	3		DENSITY	100.	YVEL	25.	SIE	3.7	+
+	NAMEi	VALUEi	-etc.-						
+	XVEL	3.5							

Field	Contents	Type	Default
TSID	Unique TICVAL number referenced from a TICEUL entry	I > 0	Required
METHOD	RADIAL: initializes material with radial profiles. The entries VALUEi are interpreted as table IDs. See Remarks 5, 6, and 7.	C	
NAMEi	Variable to be initialized. See *Dytran User's Guide*, Chapter 9: Running the Analysis, Result Types	C	Required
VALUEi	Value of the variable	R	Required

Remarks

1. Initial conditions for geometric regions that are not specified on TICVAL entries are assumed to be zero except density, which is set to the reference density.
2. For the Euler solvers, one can, in addition to the "normal" element variables that the solver has defined, also define an initial radial velocity field. You have to enter the location of the center from where the radial emerges, the velocity to be applied to the element center, and the decay coefficient. The center is defined by the keywords X-CENTER, Y-CENTER, Z-CENTER, the radial velocity by R-VEL and the decay coefficient by DECAY. You have to input these keywords in the above order, and have every keyword followed by its value. For more information, please refer to the *Dytran User's Guide*, Chapter 3: Constraints and Loading, Eulerian Loading and Constraints. Note that the dimension of RVEL changes with the value of DECAY. For example when DECAY equals 2, the dimension of RVEL is 1/length^2.
3. Note that the initialization of geometric regions is based on the fraction of the element that lies within the region. When, for example, an element is only partly covered by the geometric region, the initialization will be done according to the mass that lies within the defined region.
4. As many continuation lines as required can be used to specify all the variables to be initialized. A blank field terminates the list.

5. TYPE = RADIAL allows to map results of a spherical symmetric 1-D solution onto a full 3-D model. For each initialized variable, a 1-D table has to be defined that specifies the variable value for a number of distances from the center. The center is by default (0,0,0) but can be changed by setting X-CENTER, Y-CENTER, Z-CENTER. The velocity is a radial velocity and has to be specified as R-VEL. Its value is a TABLE ID.
6. PARAM SPHERSYM can be used to define a proper 1-D spherical mesh and speeds up the run by taking only the mesh-size in radial direction into account. With Patran, radial profiles can be created and written out to .xyd files. These files can be used to define tables by the TABFILE entry. For details refer to Examples 3.4 and 3.5.
7. Radial initialization of JWL is supported. The entries DESTPH and the JWL entry from the 1-D spherical solution stage have to be included in the remap run. Alternatively, the 1-D solution may be run with JWL and the follow-up run with ideal gas, provided that all JWL material has fully ignited. Radial initialization of EOSIG is not supported. In the follow-up run, ideal gas material has to be used instead of IG material.

TLOAD1

Transient Dynamic Load

Defines a transient dynamic load, enforced motion, or an Eulerian boundary condition.

Format and Example

1	2	3	4	5	6	7	8	9	10
TLOAD1	SID	LID		TYPE	TID				
TLOAD1	5	7			13				

Field	Contents		Type	Default
SID	Load number.		I > 0	Required
LID	Number of a set of loads (DAREA, FORCEn, RFORCE, MOMENT, GRAV, PLOADn, FLOW, FLOWEX, MOMENT) that defines the loading type, position, and value.		I > 0	Required
TYPE	Nature of the dynamic excitation:		I	0
	0	Force on a grid point		
		Pressure on a Lagrangian element		
		GRAV applied to model		
		RFORCE applied to model		
		ATBACC applied to ATB segments		
	2	Velocity of a Lagrangian or ALE (Eulerian) grid point		
	4	FLOW boundary condition		
	12	Velocity of a rigid body		
	13	Force on a rigid body		
TID	Number of a TABLED1 or TABLEEX entry defining the variation of load with time or by means of a user routine. If blank or zero, the loads do not vary with time.		I ≥ 0	No time variation

Remarks

1. See the FORCEn, MOMENTn, DAREA, PLOADn, GRAV, RFORCE,ATBACC FLOW, FORCEEX, and FLOWEX, entries for a description of how the loading or motion is calculated.
2. There can be one or more TLOAD1 entries in a set.
3. Transient loads to be used by Dytran must be selected in the Case Control Section (TLOAD = SID).
4. TID must be blank if it references the FLOW, FLOWEX, FLOWC, FLOWCDR, FLOWCSQ, FLOWDIR, FLOWSQ, FLOWT, FLOWTSQ, FLOWXDR, or FLOWXSQ entries.

5. If TYPE is 0, the LID field can reference any of the entries: FORCEn, MOMENTn, GRAV, RFORCE, DAREA, or PLOADn and apply the appropriate type of load.

   If TYPE is 2, the LID field can only reference DAREA, FORCE, MOMENT, FORCE3, or FORCEEX entries and applies enforced velocity to the specified grid points.

   If TYPE is 4, the LID field can only reference FLOW, FLOWEX, FLOWC, FLOWCDR, FLOWCSQ, FLOWDIR, FLOWSQ, FLOWT, FLOWTSQ, FLOWXDR, or FLOWXSQ entries and applies a flow boundary condition to the specified Eulerian faces. Be careful that FLOWDEF does not require TLOAD1 reference.

   If TYPE is 12, the LID field can only reference the DAREA, FORCE, or MOMENT entries and applies an enforced velocity to the center of the specified rigid body.

   If TYPE is 13, the LID field can only reference the FORCE or MOMENT entries and applies a force or moment to the center of the specified rigid body.

6. If more than one velocity boundary condition (TYPE = 2) is applied to a grid point, the boundary conditions can only be merged when the boundary conditions are consistently defined.

TLOAD2 Transient Dynamic Load, Form 2

Defines a transient dynamic load or enforced motion of the following form:

$$Y(t) = 0 \text{ for } \dot{t} < 0 \text{ or } \dot{t} > T_2 - T_1$$

$$Y(t) = A\dot{t}^B e^{c\dot{t}} \cos\left(2\pi F t + \frac{\pi P}{180}\right) \text{ for } 0 \le \dot{t} \le T_2 - T_1$$

where $\dot{t} = t - T_1$, and t is the analysis time.

Format and Example

1	2	3	4	5	6	7	8	9	10
TLOAD2	SID	LID		TYPE	T1	T2	F	P	+
TLOAD2	5	7		2	0.	10.E-3	1000.	90.	+
+	C	B							
+	.0	2.							

Field	Contents		Type	Default
SID	Set number		I > 0	Required
LID	Number of a set of loads (DAREA, FORCE, MOMENT, PLOAD, GRAV, RFORCE) that defines the loading type, position, and scale factor A		I > 0	Required
TYPE	Nature of the dynamic excitation:		I	0
	0	Force on a grid point		
		Pressure on a Lagrangian element		
		GRAV applied to model		
		RFORCE applied to model		
		ATBACC applied to ATB segments		
	2	Velocity of a Lagrangian grid point		
	12	Velocity of a rigid body		
	13	Force on a rigid body		
T1	Time constant		R	0.0
T2	Time constant (T2 > T1)		R	0.0
F	Frequency in cycles per unit time		R ≥ 0.0	0.0
P	Phase angle in degrees		R	0.0

Field	Contents	Type	Default
C	Exponential coefficient	R	0.0
B	Growth coefficient	R	0.0

Remarks

1. See the FORCEn, MOMENTn, DAREA, PLOADn, GRAV, RFORCE, ATBACC, FLOW, FORCEEX, and FLOWEX entries for a description of how the loading or motion is calculated.
2. There can be one or more TLOAD1 and TLOAD2 entries in a set.
3. Select transient loads to be used by Dytran in the Case Control Section (TLOAD = SID).
4. If TYPE is 0, the LID field can reference any of the entries: FORCEn, MOMENTn, DAREA, PLOAD, GRAV, or RFORCE and applies the appropriate type of load.

   If TYPE is 2 or 3, the LID field can only reference DAREA, FORCE, MOMENT, or FORCEEX entries and applies enforced velocity or acceleration to the specified grid points.

   If TYPE is 12, the LID field can only reference the DAREA, FORCE, or MOMENT entries and applies an enforced motion to the center of gravity of the specified rigid bodies.

   If TYPE is 13, the LID field can only reference the FORCE or MOMENT entries and applies a force or moment to the center of the specified rigid body.
5. If more than one velocity boundary condition (TYPE = 2) is applied to a grid point, the boundary conditions are constant velocity boundary conditions and are consistently defined.

VISCDMP

Dynamic Relaxation

Defines the dynamic relaxation for the various types of Lagrangian elements, rigid bodies, and ellipsoids.

Format and Example

1	2	3	4	5	6	7	8	9	10
VISCDMP	SOLST	SOLEND		SOLV1			SOLTBL1		+
VISCDMP									+
+	SHST	SHEND		SHV1			SHTBL1		+
+									+
+	MEMST	MEMEND		MEMV1		MEMV2	MEMTBL1	MEMTBL2	+
+									+
+	EL1DST	EL1DEND		EL1DV1			EL1DTBL1		+
+	0.	10.E-3		0.01					+
+	RIGST	RIGEND		RIGV1			RIGTBL1		+
+	0.	10.E-3		0.05					+
+	ELLST	ELLEND		ELLV1			ELLTBL1		
+	0.	10.E-3		0.06					

Field	Contents	Type	Default
SOLST	Start time for solid-element dynamic relaxation	R ≥ 0	0.0
SOLEND	End time for solid-element dynamic relaxation	R ≥ 0	1.E20
SOLV1	Dynamic relaxation factor for grid points of solid elements	R ≥ 0	0.0
SOLTBL1	Number of a TABLED1 entry giving the variation of dynamic relaxation for grid points of solid elements (y-value) with time (x-value).	I>0	See Remarks 2. and 3.
SHST	Start time for shell-element dynamic relaxation	R ≥ 0	0.0
SHEND	End time for shell-element dynamic relaxation	R ≥ 0	1.E20
SHV1	Dynamic relaxation factor for grid points of shell elements	R ≥ 0	0.0
SHTBL1	Number of a TABLED1 entry giving the variation of dynamic relaxation for grid points of shell elements (y-value) with time (x-value).	I>0	See Remarks 2. and 3.

Field	Contents	Type	Default
MEMST	Start time for membrane-element dynamic relaxation	R ≥ 0	0.0
MEMEND	End time for membrane-element dynamic relaxation	R ≥ 0	1.E20
MEMV1	Dynamic relaxation factor for grid points of membrane elements	R ≥ 0	0.0
MEMV2	Dynamic relaxation factor for membrane element stiffness	R ≥ 0	0.0
MEMTBL1	Number of a TABLED1 entry giving the variation of dynamic relaxation for grid points of membrane elements (y-value) with time (x-value).	I>0	See Remarks 2. and 3.
MEMTBL2	Number of a TABLED1 entry giving the variation of dynamic relaxation for membrane element stiffness (y-value) with time (x-value).	I>0	See Remarks 2. and 3.
EL1DST	Start time for one-dimensional element dynamic relaxation	R ≥ 0	0.0
EL1DEND	End time for one-dimensional element dynamic relaxation	R ≥ 0	1.E20
EL1DV1	Dynamic relaxation factor for grid points of one-dimensional elements	R ≥ 0	0.0
EL1DTBL1	Number of a TABLED1 entry giving the variation of dynamic relaxation for grid points of one-dimensional elements (y-value) with time (x-value).	I>0	See Remarks 2. and 3.
RIGST	Start time for rigid-body dynamic relaxation	R ≥ 0	0.0
RIGEND	End time for rigid-body dynamic relaxation	R ≥ 0	1.E20
RIGV1	Dynamic relaxation factor for the rigid-body masses	R ≥ 0	0.0
RIGTBL1	Number of a TABLED1 entry giving the variation of dynamic relaxation for the rigid-body masses (y-value) with time (x-value).	I>0	See Remarks 2. and 3.
ELLST	Start time for ellipsoid dynamic relaxation	R ≥ 0	0.0
ELLEND	End time for ellipsoid dynamic relaxation	R ≥ 0	1.E20
ELLV1	Dynamic relaxation factor for the ellipsoid masses	R ≥ 0	0.0
ELLTBL1	Number of a TABLED1 entry giving the variation of dynamic relaxation for the ellipsoid masses (y-value) with time (x-value).	I>0	See Remarks 2. and 3.

Remarks

1. A dynamic relaxation factor defined for a certain element type applies to all elements of that type present in the problem.
2. Dynamic relaxation factor can be defined by providing either start time, end time, and a constant factor or a TABLED1 number defining dynamic relaxation factor vs time. Providing both will raise an error.

3. Time-varying dynamic relaxation factor can be defined using TABLED1 entry. Example:

```
$  01   ><   02   ><   03   ><   04   ><   05   ><   06   ><   07   ><   08   ><   09   >< 10    >
VISCDMP                                                                                 +
+                                                                    100                +
+                                                                                       +
+                                                                                       +
+                                                                                       +
+
$  01   ><   02   ><   03   ><   04   ><   05   ><   06   ><   07   ><   08   ><   09   >< 10    >
TABLED1       100                                                                       +
$                     TIME    VISCDMP_VALUE
*                     0.00             0.00                                           *
*                     0.02             0.00                                           *
*                     0.02           1.0E-3                                           *
*                     0.03           1.0E-3                                           *
*                     0.03             0.00                                           *
*                     0.05             0.00                                           *
*                     0.05           1.0E-2                                           *
*                     0.06           1.0E-2                                           *
*                     0.06             0.00                                           *
*                   1.0E20             0.00     ENDT
$
```

4. The dynamic relaxation provided for Lagrangian CHEXA, CQUAD4, and CTRIA3 elements will be overwritten by NASINIT entry, except when the dynamic relaxation is defined using TABLED1 entry.
5. See also *Dytran Theory Manual*, Chapter 4: Models, Dynamic Relaxation for general information on dynamic relaxation in Dytran.

WALL

Lagrangian Rigid Wall

Defines a rigid plane through which specified Lagrangian grid points cannot penetrate.

Format and Example

1	2	3	4	5	6	7	8	9	10
WALL	ID	XP	YP	ZP	NX	NY	NZ	SET	+
WALL	17						1.0	21	
+	METHOD	FS	FK	EXP					
	PENALTY	0.2							

Field	Contents		Type	Default
ID	Unique rigid-wall number		I > 0	Required
XP, YP, ZP	Coordinates of the origin of the wall		R	0.0
NX, NY, NZ	A vector normal to the wall pointing towards the model		R	0.0
SET	Number of a SET1 entry listing the points that cannot penetrate the wall.		I > 0	Required
METHOD	Algorithm for contact processing:		C	PENALTY
	PENALTY	penalty method, allowing for extra boundary conditions, friction and output.		
	KINMATIC	kinematic method, only included for compatibility reasons with older versions. This method allows no extra boundary conditions, no friction and no output.		
FS	Static coefficient of friction (see Remark 5.)		R ≥ 0	0.0
FK	Kinetic coefficient of friction (see Remark 5.)		R ≥ 0	0.0
EXP	Exponential decay coefficient (see Remark 5.)		R ≥ 0	0.0

Remarks

1. A rigid plane of infinite size is generated that the grid points cannot penetrate. The plane is fixed in space.
2. The grid points can slide on the wall and separate from it.
3. A (moving) rigid plane of finite size can be modeled by using a rigid surface and a primary-secondary contact.

4. For the wall definition using penalty method, output can be requested by referencing it in a SET command in the Case Control Section. The keywords for output are WALLS and WALLOUT.
5. The coefficient of friction is given by:

$$\mu = \mu_k + (\mu_s - \mu_k)e^{-\beta v}$$

where

μ_s	=	static coefficient of friction FS
μ_k	=	kinetic coefficient of friction FK
β	=	exponential decay coefficient EXP
v	=	relative sliding velocity at the point of contact

WALLDIR

Wall Boundary Condition for all Eulerian Boundary Faces in a Specific Direction

Defines a barrier for transport in an Eulerian mesh. The boundary consists of all Eulerian boundary faces that point in a specific direction. Useful to define a barrier when a FLOWDEF has been defined. Can be used to model a floor in blast wave simulations.

Format and Example

1	2	3	4	5	6	7	8	9	10
WALLDIR	WID	ELTYPE	MESH	DIRECTION	SKFRIC				
WALLDIR	120	HYDRO		NEGX					

Field	Contents	Type	Default
WID	Unique WALLDIR number	I > 0	Required
ELTYPE	The element type to which the boundary conditions have to be applied. Allowed values are: HYDRO, MMHYDRO, and MMSTREN.	C	Required
MESH	Denotes the ID of the Euler mesh to which the boundary condition has to be applied.	I > 0	See Remark 5.
DIRECTION	Allowed values are NEGX, POSX, NEGY, POSY, NEGZ, and POSZ.	C	Required

Remarks

1. WALLDIR can be used to specify flow boundaries for CHEXA's and also for Euler element created by the MESH,BOX option.
2. In the OUT file, the total area of boundary faces is printed.
3. WALLDIR is not supported by the single material strength Euler solver.
4. WALLDIR overrules FLOW and WALLET definitions, but FLOWSQ overrules WALLDIR.
5. The MESH-ID is only used when multiple Euler domains have been defined. If multiple euler domains have been defined but if the MESH-ID is blank, all Euler domains will be considered in assigning the boundary condition.
6. The skin friction is defined as:

$$C_f = \frac{\tau_w}{0.5 \cdot \rho u^2}$$

Here, τ_w denotes the shear friction in an Euler element adjacent to the wallet, ρ is the density, and u is the tangential relative velocity in this Euler element.

WALLET

Barrier for Eulerian Transport

Defines a barrier for transport in an Eulerian mesh.

Format and Example

1	2	3	4	5	6	7	8	9	10
WALLET	WID	SID	SKFRIC						
WALLET	100	20							

Field	Contents	Type	Default
WID	Unique wall number	I > 0	Required
SID	Number of a set of CSEG, CFACE, and CFACE1 entries that define the element faces that are barriers to Eulerian transport.	I > 0	Required
SKFRIC	Skin friction value.	R ≥ 0.0	See Remark .

Remarks

1. Material cannot pass through any of the faces referenced by the SID field.
2. Barriers can be modeled on the outside as well as the inside of an Eulerian mesh.
3. See *Dytran User's Guide*, Chapter 3: Constraints and Loading, Eulerian Loading and Constraints for a more detailed description of the use of Eulerian barriers.
4. The skin friction is defined as:

$$C_f = \frac{\tau_w}{0.5 \cdot \rho u^2}$$

Here, τ_w denotes the shear friction in an Euler element adjacent to a couple surface segment where ρ is the density and u is the tangential relative velocity in this Euler element.

YLDEX

User-defined Yield Model

Specifies that a user subroutine is being used to define the yield model.

Example in FMS Section of the Dytran input stream:

```
CONNECT SERVICE myld SCA.MDSolver.Obj.Uds.Dytran.Materials
```

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDEX	YID	GROUP							
YLDEX	200	myid							

Field	Contents	Type	Default
YID	Unique yield model number.	I > 0	Required
GROUP	The group name used for the FMS section CONNECT SERVICE statement	C	Required

Remarks

1. For a description of how to use user-written subroutines, see Chapter 7: User Defined Services in this manual.
2. This yield model is applicable only for Lagrangian solid elements and Eulerian elements with shear strength.

YLDHY

Hydrodynamic Yield Model

Defines a yield model with zero yield stress.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDHY	YID								
YLDHY	200								

Field	Contents	Type	Default
YID	Unique yield-model number referenced from a DMAT entry	I > 0	Required

Remark

This yield model should be used for fluids that have no shear strength.

YLDJC

Johnson-Cook Yield Model

Defines a Johnson-Cook yield model where the yield stress is a function of effective plastic strain, strain rate, and temperature.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDJC	YID	A	B	n	C	m	EPS0	CP	+
YLDJC	100	200E6	50.E6	0.1	.15	.95	1.	285.	+
+	TMELT	TROOM							
+	1500.	273.							

Field	Contents	Type	Default
YID	Unique yield-model number referenced from a DMAT or DMATEP entry	I > 0	Required
A	Static yield stress	R ≥ 0.0	Required
B	Hardening parameter	R	0.0
n	Hardening exponent	R	1.0
C	Strain-rate parameter	R	0.0
m	Temperature exponent	R	1.0
EPS0	Reference strain rate	R > 0.0	1.0
CP	Specific heat	R > 0.0	1.E20
TMELT	Melt temperature	R	1.E20
TROOM	Room temperature	R	293.0

Remarks

1. This yield model is described in *Dytran Theory Manual*, Chapter 4: Models, Yield Models.

   The yield stress is computed from

   $$= (A + B\varepsilon_p^n)\left(1 + C ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)(1 - 1$$

   where

ε_p	=	effective plastic strain
T^*	=	$\dfrac{(T - T_r)}{(T_m - T_r)}$
$\dot{\varepsilon}$	=	effective strain rate
$\dot{\varepsilon}_0$	=	reference strain rate
T	=	temperature
T_r	=	room temperature
T_m	=	melt temperature
A, B, n, C, and m are constants.		

2. The reference strain rate is per unit time.

YLDMC

Mohr-Coulomb Yield Model

Defines a Mohr-Coulomb yield model.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDMC	YID	Y1	Y2	Y3					
YLDMC	1	10.E5	20.E5	1.E4					

Field	Contents		Type	Default
YID	Unique yield-model number referenced from:		I > 0	Required
	DMAT	for Eulerian elements with shear strength.		
Y1, Y2, Y3	Yield parameters.		R	Required
	Y1	Cap yield stress	R	Required
	Y2	Cohesion	R	Required
	Y3	Internal friction angle	R	Required

Remarks

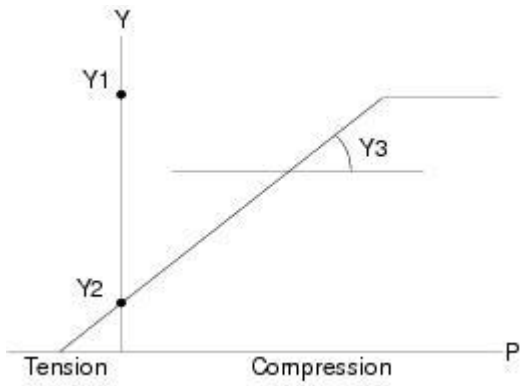
1. For a description of the yield models, see *Dytran Theory Manual*, Chapter 4: Models, Yield Models.

   The yield stress depends on the pressure as:

   $$\sigma_y = MIN(Y1, (Y2 + Y3*P))$$

   where $Y1$, $Y2$, and $Y3$ are constants and P is the pressure.

   This yield model is well suited for implementing standard geologic model like Mohr-Coulomb yield surface with a Tresa limit as shown in the following diagram.



2. This yield model is applicable only for Eulerian materials with shear strength.

YLDMSS

Multisurface Yield Model for Snow

Defines the yield model for snow material. This option must be used in combination with DMAT, EOSPOL, and SHREL.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDMSS	YID	Kc	T	Cc	Ac	Bc	Fc*	FTU	+
YLDMSS	7	0.149	82	1e-5	0.09	0.2	0.99	82	+
+	ALP0	Ds							
+	-0.37	0.0							

Field	Contents	Type	Default
YID	Unique yield model number referenced from DMAT	I > 0	Required
Kc	Parameter related to the angle of friction	R > 0.0	Required
T	Equivalent value of the snow cohesion; see Remark 5.	R > 0.0	Required
Cc	Shape of the yield surface; see remark 3	R > 0.0	Required
Ac	Hardening parameter for compression; see Remark 4.	R > 0.0	Required
Bc	Hardening parameter for compression; see Remark 4.	R > 0.0	Required
Fc*	Factor to avoid singularity; see Remark 4.	0 < R < 1	0.99
FTU	Hydrostatic tensile strength; see Remark 6.	R > 0.0	T/3
ALP0	Initial compressive volumetric plasticity strain; see Remark 4.	R < 0	Required
Ds	Softening modulus; see Remark 7.	R > 0.0	0.0

Remarks

1. This material model can be used to model Snow material.
2. In addition to deviatoric plastic strain there is also volumetric plastic strain. This volumetric strain is stored in the VOLPLS output variable. The deviatoric strain is stored in the EFFPLS variable.
3. For detail description of each parameter in this model, the user should refer to the Theory Manual in which the mechanical properties of snow are described.
4. If Cc is set to zero, then the material behaves as a Drucker-Prager model. Ac, Bc, Fc*, and ALP0 are ignored.
5. The T value must be consistently converted from the cohesion data, C_{DP}, of the Drucker-Prager model as follows:

$$T = C_{DP}/K_c$$

6. FTU, hydrostatic tensile strength, may not be greater than T divided by 3. Otherwise, it is set to that value.
7. The softening modulus is used to update the hardening parameter q_t; see Theory Manual. It can be requested as output using the FTU variable. The corresponding accumulated-plastic-volumetric-tensile-strain variable is SOFTE.
8. This material model is valid for the Euler with Strength processor and the Multi-material Euler with Strength solver.

YLDPOL Polynomial Yield Model

Defines a polynomial yield model where the yield stress is a function of effective plastic strain.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDPOL	YID	A	B	C	D	E	F	Smax	
YLDPOL	7	180.e6							

Field	Contents	Type	Default
YID	Unique yield model number	I > 0	Required
A	Initial yield parameter	R > 0.0	Required
B	Coefficient B	R	0.0
C	Coefficient C	R	0.0
D	Coefficient D	R	0.0
E	Coefficient E	R	0.0
F	Coefficient F	R	0.0
Smax	Maximum yield stress	R	1.E20

Remarks

1. This yield model is described in Yield Models at the beginning of this chapter.

   The yield stress is computed from

   $$\sigma_y = MIN(\sigma_{max}, A + B\varepsilon_p + C\varepsilon_p^2 + D\varepsilon_p^3 + E\varepsilon_p^4 + F\varepsilon_p^5)$$

   where

ε_p	=	effective plastic strain
σ_{max}	=	maximum yield stress
A, B, C, D, E, and F are constants.		

2. For a description of all of the yield models, see Yield Models in Chapter 4: Models of the *Dytran Theory Manual*.

YLDRPL

Rate Power Law Yield Model

Defines a rate power law yield model where the yield stress is a function of effective plastic strain and strain rate.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDRPL	YID	A	B	n	m	C			
YLDRPL	7	180.e6							

Field	Contents	Type	Default
YID	Unique yield model number	I > 0	Required
A	Initial yield parameter	R > 0.0	Required
B	Hardening parameter	R	0.0.
N	Hardening exponent	R	1.0.
M	Strain rate exponent	R	1.0.
C	Minimum yield stress	R	1.E20.

Remarks

1. This yield model is described in Yield Models at the beginning of this chapter.

   The yield stress is computed from

   $$\sigma_y = MAX(C, A + B\varepsilon_p^n \dot{\varepsilon}^m)$$

   where

ε_p	=	effective plastic strain.
$\dot{\varepsilon}$	=	effective strain rate.
A, B, n, m, and C are constants.		

2. For a description of all of the yield models, see Yield Models in Chapter 4: Models of the *Dytran Theory Manual*.

YLDSG Steinberg-Guinan Yield Model

Defines the Steinberg-Guinan yield model where the yield stress is a function of effective plastic strain, pressure and temperature.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDSG	YID	A1	A2	A3	A4	H	B	CP	+
YLDSG	7	8e+6	100e+6	110	0.5			130e+6	
+	TMELT	TROOM							
	1500	273							

Field	Contents	Type	Default
YID	Unique yield model number	I > 0	Required
A1-A4	Yield parameters	R > 0.0	Required
H, B	Yield parameters	R	0.0
CP	Specific heat	R > 0.0	1.E20
TMELT	Melt temperature	R	1.E20
TROOM	Room temperature	R	293.0

Remarks

1. This material model can be used to model metals for a wide range of strain rates.
2. This yield model is described in Yield Models in Chapter 4: Models of the *Dytran Theory Manual*.

   The yield stress is computed from:

$$A_T = A_1(1 + A_3\varepsilon_p)^{A_4}$$

$$\sigma_y = min(A_2, A_T)\left[1 - H(T - T_r) + Bp\left(\frac{\rho}{p_{ref}}\right)^{1/3}\right], T < T_m$$

$$\sigma_y = 0, T \geq T_m$$

   where

ε_p	=	effective plastic strain
T	=	temperature

T_r	=	room temperature
T_m	=	melt temperature
p	=	pressure
ρ	=	density
$A_1, \ldots, A_4, H$ and B are constants.		

3. The reference and quasi-static strain rate are per unit time.
4. For a description of all of the yield models, see Yield Models at the beginning of this chapter.

YLDTM

Tanimura-Mimura Yield Model

Defines the Tanimura-Mimura yield model where the yield stress is a function of effective plastic strain, strain rate and temperature.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDTM	YID	A	B	C	D	m	EPSM	CP	+
YLDTM	7	45.6e6	19.5e6			1.0	0.001		+
+	TMELT	TROOM	Scr	E	k	EPS0			
+			4000.e6	2100.	0.5	1.0			

Field	Contents	Type	Default
YID	Unique yield model number	I > 0	Required
A	Static yield parameter	R > 0.0	Required
B	Hardening parameter	R	0.0
C	Strain rate parameter C	R	0.0
D	Strain rate parameter D	R	0.0
m	Temperature exponent	R	1.0
EPSM	Quasi-static strain rate	R > 0.0	1.0
CP	Specific heat	R > 0.0	1.E20
TMELT	Melt temperature	R	1.E20
TROOM	Room temperature	R	293.0
Scr	Critical yield stress	R > 0.0	1.E20
E	Strain rate parameter E	R	0.0
K	Strain rate exponent	R	1.0
EPS0	Reference strain rate	R > 0.0	1.0

Remarks

1. This material model can be used to model metals for a wide range of strain rates.
2. This yield model is described in Yield Models at the beginning of this chapter.

   The yield stress is computed from

$$4 + B\varepsilon_p + (C + D\varepsilon_p)\left(1 - \frac{A + B\varepsilon_p}{\sigma_{cr}}\right) ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_s}\right)\Bigg](1 - T^{*m}) +$$

where

ε_p	=	effective plastic strain
σ_{cr}	=	critical yield stress
$\dot{\varepsilon}$	=	effective strain rate
$\dot{\varepsilon}_s$	=	quasi-static strain rate
$\dot{\varepsilon}_0$	=	reference strain rate
T^*	=	$\frac{(T - T_r)}{(T_m - T_r)}$
T	=	temperature
T_r	=	room temperature
T_{mr}	=	melt temperature
A , B , C , D , m , E , and k are constants		

3. The reference and quasi-static strain rate are per unit time.
4. For a description of all of the yield models, see Yield Models at the beginning of this chapter.

YLDVM

von Mises Yield Model

Defines a bilinear or piecewise-linear yield model with isotropic hardening, using the von Mises yield criterion.

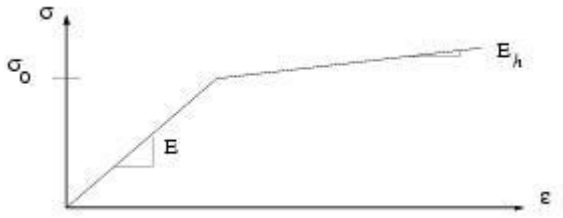
Format and Example

1	2	3	4	5	6	7	8	9	10
YLDVM	YID	YIELD	EH						+
YLDVM	32	250.E6	2000.E6						+
+	TABLE	TYPE	TABY	D	P				
+									

Field	Contents		Type	Default
YID	Unique yield-model number		I > 0	Required
YIELD	Yield stress		R	Required
EH	Hardening modulus		R	0.0
TABLE	Number of a TABLED1 entry giving the variation of effective stress (y-value) with effective strain (x-value).		I > 0	See Remark 5.
TYPE	The type of stress and strain defined in TABLED1.			TRUE
	ENG	Engineering stress and strain.		
	TRUE	True stress and strain.		
	PLAST	True stress and plastic strain.		
	PMOD	Plastic modulus and true stress.		
TABY	Number of TABLED1 entry giving the variation of the scale factor for the yield stress (y-value) with the strain rate (x-value). Strain-rate effects are also specified using the Cowper-Symonds relation (see input parameters D and P).		I > 0	See Remark 7.
D	Factor D in the Cowper-Symonds rate enhancement equation		R ≥ 0	See Remark 7.
P	Factor P in the Cowper-Symonds rate enhancement equation		R ≥ 0	See Remark 7.

Remarks

1. A bilinear stress-strain characteristic is used by specifying YIELD and EH:



where the yield stress σ_y is given by

$$\sigma_y = \sigma_0 + \frac{EE_h}{E - E_h}\varepsilon_p$$

where

σ_0	=	yield stress specified in the YIELD field
E	=	Young's modulus
E_h	=	hardening modulus specified in the EH field
ε_p	=	equivalent plastic strain
σ_y	=	yield stress

2. A piecewise linear, stress-strain characteristic is used by specifying TABLE and TYPE (beams and shells only)

$$\sigma = [(\sigma_i - \sigma_{i-1})(\varepsilon - \varepsilon_{i-1})/(\varepsilon_i - \varepsilon_{i-1})] + \sigma_{i-1}$$

The stress-strain characteristic used internally in Dytran is in terms of true stress and equivalent plastic strain. However, for convenience, the stress-strain characteristic can be input in any of the following ways (see *Dytran Theory Manual*, Chapter 4: Models, Yield Models):

True stress/true strain	(TYPE = TRUE)
Engineering stress/engineering strain	(TYPE = ENG)
True stress/plastic strain	(TYPE = PLAST)
Plastic modulus/true stress	(TYPE = PMOD)

3. For a description of all of the yield models, see *Dytran Theory Manual*, Chapter 4: Models, Yield Models.
4. With Lagrangian and Eulerian solid elements, only an elastic perfectly plastic yield model is currently used. Only the YIELD field is used.

5. If TABLE is blank or zero, a bilinear stress-strain curve is assumed. If TABLE has a value, it refers to a TABLED1 entry giving the stress-strain curve for the material.
6. If TABLE is defined, the value of YIELD is left blank, since it is determined from the stress-strain curve.
7. If TABY is blank or zero and D and P are blank or zero, the yield stress does not vary with strain rate. If TABY has a value, then it references a TABLED1 entry, which gives the variation of the scale factor applied to the yield stress with strain rate. (D and P must be blank or zero.)

   If TABY is blank or zero and D and P are defined, the enhancement of the yield stress with strain rate is calculated as

$$\frac{\sigma_d}{\sigma_y} = 1 + \left(\frac{\dot{\varepsilon}_p}{D}\right)^{1/p}$$

   where σ_d is the dynamic stress, σ_y is the static yield stress (YIELD), and $\dot{\varepsilon}_p$ is the equivalent plastic strain rate.
8. If TYPE is PLAST or PMOD, Young's modulus must be defined. If TYPE is ENG or TRUE and Young's modulus is defined it overrides the value calculated from the stress-strain curve. See *Dytran Theory Manual*, Chapter 4: Models, Yield Models for more details.
9. Note that for values exceeding the maximum x-value of either of the TABLED1 entries (see TABLE and TABY fields), linear extrapolation is used based upon the last two points specified in the TABLED1.

YLDZA

Zerilli-Armstrong Yield Model

Defines the Zerilli-Armstrong yield model where the yield stress is a function of effective plastic strain, strain rate and temperature.

Format and Example

1	2	3	4	5	6	7	8	9	10
YLDZA	YID	A	B	n	C	m	EPS0	CP	+
YLDZA	7	200.e6	50.e6	0.1					+
+	D								
+	0.0								

Field	Contents	Type	Default
YID	Unique yield model number	I > 0	Required
A	Static yield parameter	R > 0.0	Required
B	Hardening parameter	R ≥ 0.0	0.0
n	Hardening exponent	R > 0.0	1.0
C	Strain rate parameter	R	0.0
m	Temperature exponent	R	1.0
EPS0	Reference strain rate	R > 0.0	1.0
CP	Specific heat	R > 0.0	1.E20
D	Bcc parameter	R	Fcc metals

Remarks

1. This material model can be used to model to model Fcc (iron and steels) and Bcc (aluminum and alloys) metals.
2. This yield model is described in Yield Models at the beginning of this chapter.

   The yield stress for Fcc metals and Bcc metals is:

   $= (A + B\varepsilon_p^{\ N})e^{\left[-MT + CTln\right|}$ for Fcc metals.

   $(A + B\varepsilon_p^{\ N}) + De^{\left[-MT + CTln\right.}$ for Bcc metals.

where

ε_p	=	effective plastic strain
$\dot{\varepsilon}$	=	effective strain rate
$\dot{\varepsilon}_0$	=	reference strain rate
T	=	temperature
A, B, n, C, m, and D are constants.		

3. The reference strain rate are per unit time.
4. In case the Bcc parameter D is not supplied, it is assumed that a Fcc metal is defined.
5. For a description of all of the yield models, see Yield Models at the beginning of this chapter.

6 Parameters

- Parameters Summary 643
- ALEITR 649
- ALETOL 650
- ALEVER 651
- ATB-H-OUTPUT 652
- ATBAOUT 653
- ATBSEGCREATE 654
- ATBTOUT 655
- AUTHINFO 656
- AUTHQUEUE 657
- AUTOCOUP 658
- AXIALSYM 661
- AXREMAP 663
- BULKL 664
- BULKQ 665
- BULKTYP 666
- CFULLRIG 667
- CLUFLIM 668
- CLUMPENER 669
- COHESION 670
- COMPSYS 671
- COMPTCK 672
- CONM2OUT 673
- CONTACT 674
- COSUBCYC 678
- COSUBMAX 679
- COSUBMAXT 680
- COUFRIC 681
- DELCLUMP 682
- DMPOPT 683

- DUMSEGS 684
- ELDLTH 685
- ERRUSR 686
- EULER-BOUNDARY 687
- EULERCUB 688
- EULERPR 689
- EULSTRESS 691
- EULTRAN 692
- EUSUBCYC 693
- EUSUBMAX 694
- FAILDT 696
- FAILOUT 697
- FALENEG 698
- FASTCOUP 699
- FBLEND 702
- FLOW-METHOD 703
- FMULTI 706
- GEOCHECK 707
- GRADED-MESH 708
- HGCMEM 709
- HGCOEFF 710
- HGCSOL 711
- HGCTWS 712
- HGCWRP 713
- HGSHELL 714
- HGSOLID 715
- HGTYPE 716

- HICGRAV 717
- HVLFAIL 718
- HYDROBOD 719
- IEEE 720
- IGNFRCER 721
- IMM 722
- INFO-BJOIN 724
- INISTEP 725
- INITFILE 726
- INITNAS 728
- JWLDET 730
- LAGPR 731
- LIMCUB 732
- LIMITER 733
- MATRMERG 734
- MATRMRG1 735
- MAXSTEP 736
- MESHELL 737
- MESHPLN 738
- MICRO 739
- MINSTEP 740
- MIXGAS 741
- NASIGN 742
- NZEROVEL 743
- OLDLAGTET 744
- OLDSHPL 745
- PARALLEL 746

- PLCOVCUT 747
- PMINFAIL 748
- RBE2INFO 749
- RHOCUT 750
- RJSTIFF 751
- RKSCHEME 752
- ROHYDRO 753
- ROMULTI 754
- ROSTR 755
- RSTDROP 756
- SCALEMAS 757
- SHELLFORM 759
- SHELMSYS 761
- SHPLAST 762
- SHSTRDEF 763
- SHTHICK 764
- SLELM 765SMF 766
- SMF 766
- SMP, ,BATCHSIZE 768
- SMP, ,CPUINFO 770
- SMSOPT 771
- SNDLIM 773
- SPREMAP 774
- SPHERSYM 775
- STEPFCT 776
- STEPFCTL 777
- STRNOUT 778
- TOLCHK 779
- TOLFAC 780
- UGASC 781
- VARACTIV 782

- VDAMP 784
- VELCUT 785
- VELMAX 786
- VELMAX1 787
- VISCOPLAS 789

Overview

The PARAM entry in the Bulk Data Section of the input file is used to change a number of the values that control the analysis. This section describes all possible options and values that can be set with the PARAM entry.

All the options are set to default values if no PARAM entry with that option is present in the input file. You therefore only need to include a PARAM entry if you want to change one of the defaults. Normally, the default values work well and need not be altered.

Parameters Summary

The following PARAM entries are available:

Contact Control

CONTACT	Sets the contact defaults
LIMCUB	Contact cube-sort algorithm

Coupling Subcycling

COSUBCYC	Growth of subcycling in coupling
COSUBMAX	Subcycle limit

Blending Control

DELCLUMP	Delete clump factor
FBLEND	Blend fraction
CLUMPENER	Kinetic energy calculation method for Eulerian blended clumps

Time-Step Control

INISTEP	Initial time step
MAXSTEP	Maximum time step
MINSTEP	Minimum time step.
SCALEMAS	Mass scaling definition
STEPFCT	Time-step scale factor

Limits

FMULTI	Dimensioning of multimaterial overflow array
MICRO	Microzoning parameter
RHOCUT	Global minimum density for Eulerian elements
ROHYDRO	Minimum density for hydrodynamic, single-material Eulerian elements
ROMULTI	Minimum density for multimaterial Eulerian elements
ROSTR	Minimum density for single material Eulerian elements with strength
SNDLIM	Minimum value of speed of sound
VELCUT	Velocity cutoff
VELMAX	Maximum velocity for Eulerian and Lagrangian elements

Restart Control

RSTDROP	Type of elements dropped when restarting

ALE Motion Control

ALEITR	Number of ALE grid iterations
ALETOL	Tolerance at ALE interface
ALEVER	ALE Volume Computation Method

Hourglass Suppression Control

HGCMEM	Shell membrane hourglass damping coefficient
HGCOEFF	Hourglass damping coefficient
HGCSOL	Solid hourglass damping coefficient
HGCTWS	Shell twisting hourglass damping coefficient
HGCWRP	Shell warping hourglass damping coefficient
HGSHELL	Shell hourglass suppression method
HGSOLID	Solid hourglass suppression method
HGTYPE	Global hourglass suppression method

Miscellaneous

AUTHQUEUE	Licensing queuing control
CFULLRIG	Converts 123456 constraints to `FULLRIG` for RBE2

EULTRAN	Switch for the multimaterial Euler transport scheme
FASTCOUP	Fast coupling algorithm
GEOCHECK	Defines geometry consistency check
HYDROBOD	Defines a body force for single hydro material in Euler
LIMITER	Defines type of scheme used in the Euler solver
MATRMERG	Merges MATRIG and RBE2FULLRIG assemblies
MATRMRG1	Merges MATRIG and RBE2 FULLRIG assemblies
RJSTIFF	Rigid-joint stiffness
RKSCHEME	Defines the type of time scheme used in the Euler solver
VARACTIV	Activation or deactivation of grid-point, element, or face variables

Material Parameter Control

BULKL	Linear bulk-viscosity coefficient
BULKQ	Quadratic bulk-viscosity coefficient
BULKTYP	Bulk-viscosity type
HVLFAIL	Switches failure at hydrodynamic volume limit
PMINFAIL	Switches failure at spall limit

Shell Options

SHELMSYS	Shell-element system definition
SHPLAST	Type of plane-stress plasticity for shells
SHTHICK	Shell-thickness modification option
SHELLFORM	Sets the default shell formulation

Dynamic Relaxation

VDAMP	Defines dynamic-relaxation parameter

Airbag Control

PLCOVCUT	Defines time that PLCOVER will be cut
UGASC	Universal gas constant

ATB Positioning

ATBSEGCREATE	Creates grids and elements for ATBSEG

Output Control

ATB-H-OUTPUT	Writes ATB output to Dytran time-history files
ATBAOUT	Frequency of output written to main output file of ATB
ATBTOUT	Frequency of output written to the time-history files of ATB
AUTHINFO	Licensing information control
CONM2OUT	Conm2 summary output
ELDLTH	Show list of Lagrangian elements with time step in ascending order
ERRUSR	Error message redefinition
FAILOUT	Failed element output parameter
IEEE	IEEE binary data output format
IGNFRCER	Ignores warnings
INFO-BJOIN	Lists the generated BJOIN s and spot welds
MESHELL	Mesh density for covering rigid ellipsoids
MESHPLN	Mesh density for covering of rigid planes
NASIGN	Echo ignored data entries
PARALLEL	Parallel execution information
RBE2INFO	Lists MATRIG and RBE2 grids in the output file
SHSTRDEF	Composite shell stress and strain output
SLELM	Store shell sublayer variables
STRNOUT	Shell sublayer strain output

Prestressing Analysis

INITFILE	Defines method of initialization from a solution file
INITNAS	Defines the type of displacement initialization file

Parallelization

DMPOPT	Sets DMP option
EULERCUB	Cube Sorting for Euler Domains
EULERPR	Euler Cube Partition Control
LAGPR	Sets METIS partitioner

Parameter Descriptions

The parameters are listed in alphabetical order. The entry and the examples are shown in free format, although small or large fixed format entries can also be used.

The default value indicates the value used if no PARAM entry is present. The type column indicates the type of data you must supply. This can be `I` (Integer), `R` (Real), or `C` (Character). In addition, a range of permissible values may also be indicated for example, `I > 0` means that you must supply an integer that is greater than zero.

ALEITR

Number of ALE Mesh Iterations

Defines the number of mesh iterations to be used to move the free ALE grid points.

Format and Example	Default
PARAM,ALEITR,value	1
PARAM,ALEITR,3	

Option	Meaning	Type
value	Number of mesh iterations	1 < I < 6

Remarks

1. In most applications, one iteration should be sufficient. If not, the number of iterations can be increased to a maximum of six.
2. Less mesh iterations reduce the computational effort.

ALETOL

Tolerance at ALE Interface

Defines the tolerance in matching Eulerian and Lagrangian grid points in the ALE interface surface

Format and Example	Default
`PARAM,ALETOL,value`	1.E–4
`PARAM,ALETOL,1.E-4`	

Option	Meaning	Type
`value`	Tolerance	R > 0.0

Remark

Grid points in the ALE interface with coordinates that fall within the tolerance are recognized to be an ALE interface pair.

ALEVER

ALE Volume Computation Method

Defines the method to be used in the element-volume computation in ALE meshes.

Format and Example	Default
`PARAM,ALEVER,option`	V2.1
`PARAM,ALEVER,V2.2`	

Option	Meaning	Type
`V2.1`	V2.1 method uses a fast approximation scheme for the element-volume computation.	C
`V2.2`	V2.2 method gives the exact element volume.	

Remark

The V2.2 option should be used in problems where the pressure levels are expected to be low. The V2.1 method is faster and consumes less CPU time, but it can lead to spurious pressure levels in a low pressure level calculation.

ATB-H-OUTPUT

Write ATB Output to Dytran Time-History Files

An Dytran time-history file is created containing the output as requested in the ATB input file on cards H.1 to H.11.

Format and Example
`PARAM,ATB-H-OUTPUT,option`
`PARAM,ATB-H-OUTPUT,NO`

Option	Meaning	Type	Default
YES	The Dytran time-history files are created.	C	YES
NO	The Dytran time-history files are not created.		

ATBAOUT Output Frequency to Main Output File of ATB

Defines the frequency at which output is written to the main output file of ATB.

Format and Example	Default
`PARAM,ATBAOUT,value`	10.0E-3
`PARAM,ATBAOUT,5.0E-3`	

Option	Meaning	Type
`value`	Every multiple of `ATBAOUT` seconds, the main output file of ATB is updated.	R > 0.0

Remarks

1. Only active when field 3 on the A5 card of the ATB input file is set to a value of -1.
2. Controls the frequency of the output of segment acceleration, velocity and displacement, joint forces and moments.

ATBSEGCREATE

Create Grids and Elements for ATBSEG

A Bulk Data file is created containing grid points and elements visualizing the initial position and orientation of the coordinate systems of the ATB segment and its joints.

Format and Example
`PARAM,ATBSEGCREATE,option,NAME,Length1,length2`
`PARAM,ATBSEGCREATE,YES,HYBRID-III,0.1,0.05`

Option	Meaning	Type	Default
`YES`	If `EID1` through `EID3` on the ATBSEG entry and/or `EID1` through `EID6` on the ATBJNT entry are defined, Dytran extracts the initial positions from the ATB input file for the grid points `G0` through `G3` from the ATBSEG entry and/or for the grid points `G0` through `G6` from the ATBSEG entry. Bulk Data entries as specified on the ATBSEG and ATBJNT entries are written to the file with name `ATB_<NAME>.DAT`, where `NAME` is equal to the name specified on this s entry.		
`NO`	The specifications for `EID1` through `EID3` on the ATBSEG entry and/or `EID1` through `EID6` on the ATBJNT entry are ignored. No Bulk Data file is created.		
`NAME`	Name given to the Bulk Data file	C	Required
`LENGTH1`	Specifies the length of the axes spanned by the grid points that represent the local coordinate systems of the segments	R > 0.0	0.1
`LENGTH2`	Specifies the length of the axes spanned by the grid points that represent the local coordinate systems of the joints	R > 0.0	0.05

ATBTOUT — Output Frequency to Time-History Files of ATB

Defines the frequency at which output is written to the time-history files of ATB.

Format and Example	Default
`PARAM,ATBTOUT,value`	1.0E-3
`PARAM,ATBTOUT,1.0E-4`	

Option	Meaning	Type
`value`	Every multiple of `ATBTOUT` seconds, the time-history files of ATB are updated.	R > 0.0

Remarks

1. Only active when field 26 on the A5 card of the ATB input file is set to a value of –1.
2. Controls the frequency of all output requested on the H-cards, and of the tabular time-histories that are controlled by field 18 on the A5 card of the ATB input file.

AUTHINFO

Licensing Information Control

Defines the amount of information FLEXlm licensing writes to the output file.

Format and Example	Default
`PARAM,AUTHINFO,value`	1
`PARAM,AUTHINFO,9`	

Option	Meaning	Type
`value`	The amount of licensing information that FLEXlm writes to the output file. A value of 1 provides the minimum amount of licensing information, while a value of 9 provides the maximum amount of information.	I > 0

Remark

You can use this parameter to obtain extra licensing information if a FLEXlm licensing problem is experienced. Under normal circumstances, where FLEXlm licensing is not a problem, this parameter is not used.

AUTHQUEUE

Licensing Queuing Control

Defines the queuing time of the FLEXlm licensing system.

Format and Example	Default
`PARAM,AUTHQUEUE,value`	30
`PARAM,AUTHQUEUE,600`	

Option	Meaning	Type
`value`	Specifies the time in minutes for FlexLM to wait for a seat to become available. If the seat becomes available before the specified time period, the job is allowed to continue. If not, the job is terminated. A value of 0 disables the FLEXlm queue functionality.	I > 0

Remarks

1. When a job is waiting for a seat to become available, it consumes computer resources such as memory, disk space, etc. Too many jobs waiting for licenses could have a severe impact on the system.
2. A maximum of 100 Dytran jobs can be queued.
3. If queueing is enabled, Dytran waits in the queue until the license of the desired type has been released by any other job(s) currently holding it. If queueing is disabled, Dytran searches for any next applicable free Dytran license with which the run could be started. When no more of the desired Dytran licenses are found, the job terminates. Note that a job requiring a 'basic' license could also run using a 'standard' license although that one would obtain more tokens than necessary.

AUTOCOUP — Automatic Coupling

Defines the automatic coupling algorithm.

Format and Example
`PARAM,AUTOCOUP,ACTIVE,CLEAN,DUMMY,TOL_AREA,OUTPUT,CHECK,FAIL`
`PARAM,AUTOCOUP,ON`

Option		Meaning	Type	Default
`ACTIVE`	ON	Activates the sub element approach. C [ON,OFF]		OFF
	OFF	Auto coupling is turned off		
`CLEAN`	ON	Cleans obsolete sub elements C [ON,OFF]		OFF
	OFF	Obsolete elements will not be removed from memory.		
`DUMMY`	ON	Puts dummy segments on the coupling surface archive [ON,OFF]		OFF
	OFF	Does not put dummy segments on the coupling surface archive.		
`TOL_AREA`	.	If the area of a hole is smaller than TOL_AREA a 1D flow computation method is used.		
`OUTPUT`		Determines how results of Euler elements that are intersected by the structure are written to the Euler archive.		
	ZERO	Write zero for the results of these intersected elements.		ZERO
	AVERAGE	Average across sub elements. Each sub element has the same weight.		
	AVEREAGEU	Average across sub element using the uncover fraction as weights.		
	MAXUNC	The results are taken from the sub element with the largest volume uncover fraction.		

Option		Meaning	Type	Default
	COVER	Uses the cover field of COUPLE to select the proper sub element. Only supported for COVER=OUTSIDE or COVER=INSIDE. Not supported for COVER =NONE.		
CHECK		Checks whether all segments of the coupling surface are fully inside the Euler elements.		NOCHKEUL
	CHEKEUL	Activates checking.		
	NOCHKEUL	No checking.		
FAIL		Activates interactive failure C [POROUS, REMOVE].		POROUS
	POROUS	Makes failed segment porous. This is the fast coupling approach.		
	REMOVE	Removes the failed segments from the coupling surface.		

Remarks:

1. Features that are not supported are:
   - Multiple Euler domain
   - Adaptive Euler meshes
   - COUPLE with option AIRBAG
   - Graded meshes
   - Euler import
   - Viscosity
   - Porosity
   - Option FAIL=POROUS and REMOVE are not supported by DMP.
   - Markers
2. With automatic coupling fluid can be defined on both sides of the coupling surface. To do this option COVER of the COUPLE entry has to be set to NONE. When using COVER = OUTSIDE fluid is only initialized in the inside region of the coupling surface. For COVER = INSIDE the opposite applies. When the coupling surface is not closed, only COVER=NONE is allowed.
3. Coupling surface failure can only defined by option FAIL.

   Defining this failure by use of COUP1FL and COUP1INT is not supported.'
4. In the auto coupling approach holes in the coupling surface are meshed with dummy segments.

For holes with large deformations PARAM DUMSEGS, ON can be used to maintain good dummy segments.

AXIALSYM

Axial Symmetric Analyses

Enables an efficient and accurate 2-D axial symmetry for Eulerian materials. A much larger time step becomes possible by not taking into account the mesh-size in circumferential direction.

Format and Example	Default
`PARAM,AXIALSYM,MESHTYPE,AXIALAXIS,SYNPLAN, PHI,ALIGN,PHI2`	
`PARAM,AXIALSYM,RECT,X,XY,2.5,YES,0`	PHI: 0 ALIG: YES PHI2: 0 The other fields have to be set.

Option	Meaning		Type
`MESHTYPE`	Two types of Euler meshes are supported:		C: [AXIAL,RECT]
	Axial symmetric meshes:	`MESHTYPE = AXIAL`	
	Rectangular meshes:	`MESHTYPE = RECT`	
`AXIAL AXIS`	X Y Z		C: [X,Y,X]
`SYMPLAN`	The approximate symmetry plane of the Euler mesh		C: [XY,YZ,ZX]
	For MESHTYPE=AXIAL:		
	It is assumed that one of the coordinate planes is an approximate symmetry plane of the Euler mesh. Although approximate symmetry is sufficient, the coordinate plane can always be made an exact symmetry plane by the use of PHI2. If for example the Euler mesh has angles 0 and 2.5, PHI2 has to be set to -1.25 to get exact symmetry.		
`PHI`	Only used for `MESHTYPE = RECT`.		R > 0.0
	Used to creates a 2-D axial symmetric mesh with angles +PHI/2 and –PHI/2		
`ALIGN`	`ALIGN` normals of oblique Euler element faces. This prevents errors in strains that can arise from small errors in Euler face normals. Only needed for `MESHTYPE=AXIAL`. For `MESHTYPE=RECT` this option is ignored.		C: [YES,NO]
`PHI2`	As a final operation rotate the mesh around the axial axis by the angle PHI2. See Remark 6.		R

Remark

1. Only available for Eulerian elements and does not support Lagrange elements. The effect of this parameter is not limited to the solvers. Also, Euler archives will reflect the modified Euler mesh geometry.
2. The Euler mesh cannot only be symmetric but can also be a rectangular mesh comprising of one layer. Using the angle specified by PHI, this Euler mesh is mapped into a 2-D axial symmetric mesh.
3. The Euler mesh has to consist of one layer.
4. Rectangular meshes that can be made 2d symmetric using the angle PHI should satisfy:
   - All boundary Euler faces are aligned with a coordinate direction
   - Only one layer thick
   - The axial symmetry axis is either on the boundary of the Euler mesh or outside the Euler mesh. It is not allowed that the axial axis is inside the Euler mesh.

   Initialization of Euler element using geometric regions as defined by the TICEUL entry is carried out onto the transformed 2d axial mesh.
5. In the time step computation the circumferential mesh-size will not be taken into account.
6. Use the PHI2 option with caution. Euler initialization is done using the mesh rotated by the angle PHI2. So after including the angle PHI2 or modifying its value, the Euler initialization should be revised.

AXREMAP 2-D Axial Symmetric Euler Archive Remap

Allows a 2-D axial symmetric Euler archive importation into a 3-D simulation.

Format and Example	Default
`PARAM,AXREMAP,X0,Y0,Z0,XN,YN,ZN, RANGE`	
`PARAM AXREMAP 0.0 0.5 0.5 1. 0. 0.`	

Option	Meaning	Type	Default
`X0, Y0, Z0`	X, Y, Z coordinate of the point at which the 2-D axial symmetric mesh is remapped.	R	0.0
`XN, YN, ZN`	Unit vector that specifies the direction of the axial axis of the 2-D axial symmetric mesh as viewed in the 3-D mesh.	R	1.0, 0.0, 0.0
`RANGE`	Only material whose distance from the axial axis is smaller than "range" will be initialized with the 2-D axi-symmetric Euler archive.	R	1e+20

Remark

1. Since 2-D axial symmetric simulations run much faster than 3-D simulation, it can save much CPU time to do the first part of the simulation with a 2-D axial symmetric mesh. Afterwards, the 2-D axial symmetric Euler archive is imported into the 3-D simulation. By default, the 2-D axial symmetric archive will not be expanded in 3-D. To enable this expansion, `PARAM,AXREMAP` has to be used. It is useful for blast wave simulations. The 2-D axial symmetric simulation has to be terminated before the blast wave approaches any 3-D structure.
2. This import of Euler archives is done by means of the `EULINIT` option.
3. To generate axial symmetric meshes `PARAM,`AXIALSYM can be used.

BULKL

Linear Bulk Viscosity Coefficient

Defines the default value of the linear bulk viscosity coefficient.

Format and Example	Default
`PARAM,BULKL,value`	0.0
`PARAM,BULKL,0.1`	

Option	Meaning	Type
`value`	Value of the linear coefficient in the bulk viscosity equation.	$R \geq 0.0$

Remarks

1. The default value works well for the majority of problems.
2. The value defined on this entry is used as the default whenever `BULKL` is blank on the `DMATxx` material entries.
3. When `BULKL` is specified on a material definition entry, the default value is overridden for that specific material.
4. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on bulk viscosity.

BULKQ

Quadratic Bulk Viscosity Coefficient

Defines the default value of the quadratic bulk viscosity coefficient.

Format and Example	Default
`PARAM,BULKQ,value`	1.0
`PARAM,BULKQ,1.6`	

Option	Meaning	Type
`value`	Value of the quadratic coefficient in the bulk viscosity equation.	$R \geq 0.0$

Remarks

1. The default value works well in the majority of situations.
2. The value defined on this entry is used as the default whenever `BULKQ` is blank on the `DMATxx` material entries.
3. When `BULKQ` is specified on a material definition entry, the default value is overridden for that specific material.
4. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on bulk viscosity.

BULKTYP

Bulk Viscosity Type

Defines the default type of bulk viscosity.

Format and Example	Default
`PARAM,BULKTYP,option`	DYNA
`PARAM,BULKTYP,DYNA`	

Option	Meaning	Type
`DYNA`	Standard DYNA3D model	C
`DYTRAN`	Enhanced DYNA model	

Remark

See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on bulk viscosity.

CFULLRIG Converts 123456 Constraints to FULLRIG on RBE2 Entries

Converts all 123456 constraints to the FULLRIG option on all entries

Format and Example	Default
PARAM,CFULLRIG,value	YES
PARAM,CFULLRIG,NO	

Option	Meaning		Type
value	YES	123456 constraints are converted to FULLRIG.	C
	NO	123456 constraints are not converted to FULLRIG.	

CLUFLIM

Limiter of Volume Stain Rate for Clumps

In some cases, airbag runs become instable. Often, this is caused by a much too large volume strain rate in a clump that consists of too many elements. These clumps typically have a small average volume uncovered fraction. The large volume strain rate causes a huge compression work and this blows up the specific internal energy. When this happens it is clearly visible in the OUT file and in the results. This PARAM activates a limiter that scales down the volume strain rate for clumps with a small average uncovered fraction. It can keep an instable airbag run stable, just like PARAM, VELMAX can keep runs stable.

Format and Example	Default
PARAM,CLUFLIM,value	(1/3)*FBLEND
PARAM,CLUFLIM,0.22	

Option	Meaning	Type
VALUE	The volume strain rate in a clump will be reduced when the average Uncovered Fraction of elements in a CLump falls below CLUFLIM. CLUFLIM has to be smaller than FBLEND. The default value of FBLEND is 0.66 giving a value of 0.22 for CLUFLIM. For more details on FBLEND refer to PARAM,FBLEND.	0<R<FBLEND

Remark

The Volume strain rate $\frac{DIV}{\Delta t}$ in clumps will be limited by

$$\left(\frac{DIV}{\Delta t}\right)_{Lim} = min\left(1, \frac{U}{CLUFLIM}\right)\left(\frac{DIV}{\Delta t}\right)$$

Here, U is the average uncovered fraction of elements in the clump as given by

$$U = \frac{\sum_{el \in clump} Uncf_{el} * Vol_{el}}{\sum_{el \in clump} Vol_{el}}$$

Here $Uncf$ and Vol denote the uncovered fraction and volume of an element inside the clump.

Therefore, only when the average uncover fraction falls below CLUFLIM, the volume strain rate is limited.

CLUMPENER Switch for Kinetic Energy Calculation Scheme of Blended Clumps

Sets the definition of the kinetic energy calculation method for Eulerian blended clumps.

Format and Example	Default
`PARAM,CLUMPENER,option`	`AVERAGE`
`PARAM,CLUMPENER,SUM`	

Option	Meaning	Type
`AVERAGE`	The kinetic energy of an Eulerian blended clump is calculated from the average velocity of the clump. The average velocity of the blended clump is computed as the sum of the momentum of each member of the clump divided by the total clump mass.	C
`SUM`	The kinetic energy of an Eulerian blended clump is calculated as the sum of the kinetic energy of each member of the clump.	

Remarks

1. With the release of Dytran 2002 R1, the default method switched from `SUM` to `AVERAGE` to minimize the possible occurrence of negative internal energy locally in Eulerian elements during the analysis.
2. For normal analysis, the default method `AVERAGE` works correctly. Only in cases of repeating an older analysis is desired, should the method `SUM` be used to obtain the same results as before.

COHESION

Cohesion for Coulomb Friction

Allows friction and sticking during tensile conditions at the coupling surface.

Format and Example	Default
`PARAM,COHESION,MAXSTRS,FRIC,REFVEL`	`NONE`
`PARAM,COHESION,8e+10,8e+5,2`	

Option	Meaning	Type
`MAXSTRS`	Maximal normal stress. Allows tensile stresses at the coupling surface as long as the normal stress does not exceed `MAXSTRS`.	R>0.0
`FRIC`	Friction stress under tensile conditions.	R>0.0
`REFVEL`	Reference value for velocity	R>0.0

Remarks

1. Only used when Coulomb friction coefficients have been specified for a `COUPLE` entry.
2. During tension any relative tangential velocity between coupling surface and Eulerian material will yield a shear stress whose magnitude equals $FRIC \cdot min\left(1, \frac{V_{rel,tangential}}{REFVEL}\right)$. This is a viscous-like friction law.
3. This shear force opposes the relative tangential movement along the coupling surface.

COMPSYS

PCOMP Shell Element System Definition

Defines the shell element system for PCOMP shells.

Format and Example	Default
PARAM,COMPSYS,option	SIDE21
PARAM,COMPSYS,MIDSIDES	

Option	Meaning		Type
Option	SIDE21	x-axis along side21	C
	MIDSIDES	x-axis connecting midpoints	

Remarks

1. SIDE21 puts the x-axis along side21 of the element, whereas MIDSIDES puts the x-axis along the vector connecting the midpoints of the side14 and side32.

COMPTCK

PCOMP Shell-Thickness Modification Option

Specifies whether or not the thickness of the shell changes with membrane straining for PCOMP.

Format and Example	Default
PARAM,COMPTCK,option	NO
PARAM,COMPTCK,YES	

Option	Meaning		Type
Option	YES	Shell thickness is modified according to the membrane strain.	C
	NO	Shell thickness is constant.	

Remarks

1. The YES option gives a true large-strain shell but requires some extra computation.
2. The location of the integration points changes with the change in the thickness of the composite element as a whole. Therefore this is an approximation and may cause inaccurate results when there are large difference of material properties under large strain values with YES option.
3. This parameter is used to invoke large strain deformation in PCOMP card for thin shell structure with biaxial orthotropic behaviour and capture necking effect for multilayer material combination of NYLON, LDPE, and BOPE.

   Also, use PARAM, COMPSYS, MIDSIDES with this option.

CONM2OUT

CONM2 Summary Output

Determines if a summary of concentrated masses and their energy and momentum is written to the output file.

Format and Example	Default
PARAM,CONM2OUT,option	YES
PARAM,CONM2OUT,NO	

Option	Meaning	Type
NO	No information about concentrated masses is written to the cycle and material summaries on the output file.	C
YES	A complete summary of concentrated masses including the associated mass, momentum, and energy is written to the output file.	C

Remark

When PARAM,CONM2OUT is set to NO, there is no summary of the concentrated mass. This means that the mass, momentum, and energy of the concentrated masses, is not added to the material and cycle summaries. Setting PARAM,CONM2OUT,NO saves memory and CPU time.

CONTACT

Sets Defaults for CONTACT

Defines certain defaults for the contact definitions.

Format and Example	Type
PARAM,CONTACT,option,value1,value2,...	Blank
PARAM,CONTACT,VERSION,V4	
PARM,CONTACT,TOLFAC	

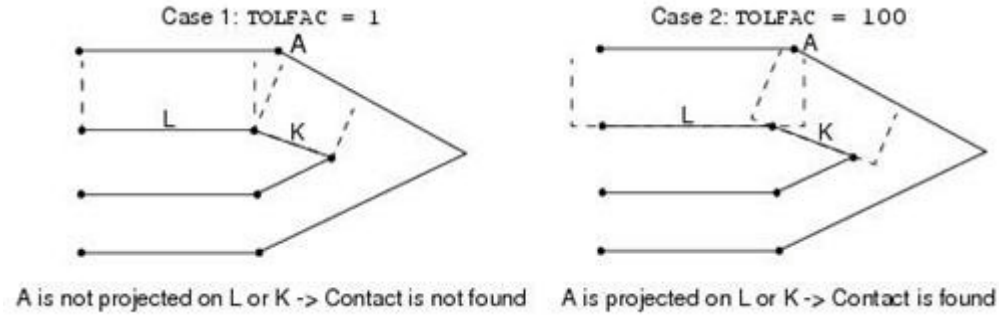
Option	Meaning		Type
VERSION,[V2,V3,V4, BELT,BELT1,DRAWBEAD]	Defines the default version.		C, Required
THICK,value	Defines the default value for THICK		C, R
GAP,value	Defines the default value for GAP		C, R > 0.0
LIMITS, [XMIN,XMAX, YMIN,YMAX, ZMIN,ZMAX]	Definition of a three dimensional contact region where contact in the analysis model takes place. Significant CPU time savings can be achieved when used in adaptive contact.		
	XMIN	Lower limit in x-direction where main contact occurs	-1.E20
	XMAX	Upper limit in x-direction where main contact occurs	1.E20
	YMIN	Lower limit in y-direction where main contact occurs	-1.E20
	YMAX	Upper limit in y-direction where main contact occurs	1.E20
	ZMIN	Lower limit in z-direction where main contact occurs	-1.E20
	ZMAX	Upper limit in z-direction where main contact occurs	1.E20
DAMPING,[YES/NO]	Defines the default setting value for DAMPING		C, C
COPOR,[YES/NO}	Activates contact based porosity. Default is NO.		C, C

Option	Meaning		Type
DAMPFOR,[YES/NO]	Defines whether the noncontact forces acting on the grid points need to be taken into account in the contact damping of the V4 contact algorithm.		C,C
	This option is only used if DAMPING is set to YES.		
	This option prevents large penetrations that might occur when the forces acting on the grid points tend to push them into the contact surface. This happens, for example, in airbag analyses, where a large pressure exists inside the bag.		
DYNA	The following parameters of the contact definition get the default values consistent with Dyna.		C
	THICK 1.0		
	GAP 0.0		
	PEN FACTOR		
	PENV 0.4		
INFO,[G1, G2, ...]	Information on the contact state of grid points G1, G2, ... is printed to ASCII files, named CNT...		C,I
	This information can be useful in debugging models with contacts.		
CONTINI, [INITIAL, ALLWAYS]	Defines how the CONTINI logic is used during the analysis. The default in INITIAL.		C,C
	INITIAL	The CONTINI logic is used only during the analysis initialization. Note that for SEARCH=SLIDE, the secondary contact nodes lose their contact once they slide off a previous page and not onto a neighbor face. This is, generally, a correct behavior for deploying airbags, except in very complicated folding patterns. For these, it is advised to use PARAM,CONTACT,CONTINI, ALLWAYS.	
	ALLWAYS	For SEARCH=SLIDE, when a secondary contact node slides off a previous face and not onto a neighbor face, new faces are checked, using the CONTINI logic.	
		For SEARCH=FULL, new faces are always checked using the CONTINI logic.	

Option	Meaning		Type
		This setting requires more memory and more CPU-time is spent on the contact logic.	
EVIEW, value	Defines the default value of the view angle of Edge to Edge contacts. The value of the angle must be in degrees.		C,R
TOLFAC, value	Scale factor for TOLPR1		C, R> 0.0 See Remarks 1. and 2.
FORCE, NMCYC, SCALE, TYPE	Controls the contact forces on the grid points.		C, 1> 0 R > 0.0, C
	NMCYC	Frequency check of contact force; applied on each grid point (Default = 100).	
	SCALE	Scale factor for maximum allowable contact force:	
		$F_{max} = SCALE * F_{last_check}$	
		(Default = 10)	
	TYPE	Contact force limitation.	
		$FMAXF\ F_{new_check}$ (where nonallowable forces are not taken into account)	
		ZERO 0.0	
		(Default = 0)	
	See Remark 3.		
STIFF, [MASS,SLSTIFF]	Defines the default contact stiffness calculation. See Remark 4.		C, Required

Remarks

1. This parameter is important for initialization of BPLANE contact. The faces of the contact surface will be enlarged with a value of TOLPR1. However, this might not be enough when the air bag is offset folded. On the other hand, a large value of TOLPR1 might induce hooking. Therefore, a new parameter is introduced called TOLFAC. The value of TOLFAC scales the value of TOLPR1 only at initialization, such that the contact is correctly found.
2. Recommended value is 100 in an air bag analysis



3. The PARAM, FORCE check takes up some CPU time and, therefore, do not make this value too small. Furthermore, when the check is performed at each cycle, the force will be too limited and the bag will not unfold. Recommended values are between 5 and 200. The same problems can occur for SCALE. In case this value is too small, the bag will not unfold either. The minimum value for air bags that should be used is about 5. The maximum is about 20. When this value is too big a difference will not be noticed. TYPE ZERO is a bigger restriction. In some cases, TYPE FMAXF might yield better results.
4. STIFF=SLSTIFF is only available with OPTION1 =ON in PARAM, SMF.

COSUBCYC

Growth of Subcycling Interval in Coupling

Controls the growth of the subcycling interval in Euler/Lagrange coupling

Format and Example	Default
`PARAM,COSUBCYC,value`	1

Option	Meaning	Type
`value`	Maximum growth of the subcycling interval.	I > 0

Remarks

1. The subcycling algorithm automatically estimates the number of subcycles to be used. This is updated throughout the calculation. This parameter controls how much the number of subcycles can grow. For example, `COSUBCYC` is set to 1, and the current number of time steps between updates of the coupling geometry is 4. If Dytran estimates that the subcycling interval should be 7, the subcycling interval is increased by 1 until a value of 7 is reached.
2. There is no control on the amount by which the subcycling interval can decrease.

COSUBMAX

Subcycle Limit in Euler/Lagrange Coupling

Defines the maximum number of subcycles that can occur in Euler/Lagrange coupling. During a subcycle, the geometry of the coupling surface is not updated.

Format and Example	Default
`PARAM,COSUBMAX,value`	1
`PARAM,COSUBMAX,10`	

Option	Meaning	Type
`value`	The maximum number of time steps between updating the coupling surface geometry in the coupling calculations.	I > 0

Remarks

1. Updating the coupling geometry takes a lot of CPU time. Subcycling gives substantial savings in CPU time for coupled calculations.
2. The smaller the value of this parameter, the greater the accuracy of the analysis and the greater the cost. Conversely, larger values offer significant CPU savings, but very large values give incorrect results.
3. If the geometry of the coupling surface is changing rapidly, smaller values of `PARAM, COSUBMAX` should be used.

COSUBMAXT Time-dependent Subcycle Limit in Euler/Lagrange Coupling

Defines the maximum number of subcycles that can occur in Euler/Lagrange coupling. During a subcycle, the geometry of the coupling surface is not updated. This number can vary in time and is given by a table.

Format and Example	Default
`PARAM,COSUBMAXT,value`	None
`PARAM,COSUBMAXT,100`	

Option	Meaning	Type
`value`	Table ID that specifies for each time the maximum number of time steps between updating the coupling surface geometry in the coupling calculations.	I >= 0

Remarks

1. Updating the coupling geometry takes a lot of CPU time. Subcycling gives substantial savings in CPU time for coupled calculations.
2. The smaller the value of this parameter, the greater the accuracy of the analysis and the greater the cost. Conversely, larger values offer significant CPU savings, but very large values give incorrect results.
3. If the geometry of the coupling surface is changing rapidly, smaller values on the table should be used.
4. It is allowed to use in the table the value zero. The coupling surface computation is done every cycle as long as the table value is zero. This allows to temporarily switch off subcycling.

COUFRIC Coupling Surface Friction for Nonmetallic Eulerian Solids

In Dytran 2005r3, the Coulomb friction scheme was updated to fully reflect friction between metals. The Coulomb friction of Dytran 2005 is more suitable for friction with nonmetals like soil and sticky fluids. This PARAM deactivates the Dytran 2005r3 scheme and uses the Dytran 2005 scheme instead. The main difference between the schemes is the treatment of tensile reconditions. In Dytran 2005r3 and beyond, a tensile condition will result in zero load on the structure part. In Dytran 2005 under tensile condition, a tensile load was applied. In addition the onset of sliding in the friction model was applied differently. In stead of using this PARAM one may also consider using PARAM,COHESION.

Format and Example	Default
PARAM,COUFRIC,value,<NO-METAL,METAL>	METAL
PARAM,COUFRIC,NO-METAL	

Option	Meaning	Type
METAL	Uses the Dytran 2005r3 scheme for Coulomb friction.	C
NO-METAL	Uses the Dytran 2005 scheme for Coulomb friction.	

Remarks

Only used when Coulomb friction coefficients have been specified for a COUPLE entry.

DELCLUMP

Delete Clump Fraction

Material in Eulerian elements of a clump with

```
fvunc < DELCLUMP * fblend
```

is eliminated. This prevents small clumps from determining the time step and prevents the leakage of small masses to isolated regions.

Format and Example	Default
`PARAM,DELCLUMP,value`	0.5
`PARAM,DELCUMP,0.1`	

Option	Meaning	Type
`value`	The value of `DELCLUMP`	$R \geq 0.0$

Remark

Also see the FBLEND parameter.

DMPOPT

Sets DMP option

Sets DMP option.

Format and Example	Default
`PARAM,DMPOPT,value1,value2,value3,value4`	1,1,0,1
`PARAM,DMPOPT,2,1`	

Option	Meaning	Type
`Value1`	Set dmp type. See remark 1. 1 Euler dmp is only activated. 2 Lagrangian dmp is only activated. 3 Both of Euler and Lagrangian dmp are activated.	I > 0
`Value2`	Set contact dmp typ. 1 Serial mode in contact 2 Parallel mode in contact (check `Value 4`)	I > 0
`Value 3`	Set number of printing summary for each item 0 Print all summaries (see remark 2)	I ≥ 0
`Value 4`	Set parallel mode type in contact -1 use the same contact domain partitioning as the element domain partitioning and all primary contact surfaces must be made by `MATRIG` definition (see remark 3)	I

Remark

1. If a user wants to set partitioning manually, Please check PARAM, EULERPR for Euler partitioning and PARAM, LAGPR for Lagrangian partitioning.
2. Since sometimes too large printing out summaries of each item causes unexpected failure of job under certain condition, this option can limit to certain number of summary information.
3. When the same contact domain partitioning as the element domain partitioning is used, the dmp performance of contact may not be enhanced. However when all deformable grid points are assigned in secondary contact body and they are involved to contact at the similar time, the dmp performance of contact will be enhanced as expected. For example, stamping of sheet metal may show a good performance.

DUMSEGS — Creating dummy segments for the auto coupling approach

Specifies how holes in the coupling surface are meshed. It is only used for the auto coupling approach. This approach is activated by PARAM, AUTOCOUP.

Format and Example
PARAM,DUMSEGS,Internal
PARAM,DUMSEGS,[ON/OFF]

Option	Meaning	Type	Default
INTERNAL	Meshes internal holes with a triangular dummy segment mesh.	C	OFF

Remarks:

1. With the auto coupling method, holes are partially or completely meshed with dummy segments. With option OFF the standard method of creating dummy segment is used. For details see Automatic Coupling (Ch. 4) in the *Dytran User's Guide*. In this standard method holes are partially meshed with dummy segments. For holes that rotate, have arbitrary form or show large deformations the standard method of creating dummy segments can cause the simulation to terminate with messages like

   ```
   %E-P4307807-P4_SUBELM_MESH_CURVED_HOLES, , ,(13),
   DUMMY SURFACE CANNOT BE TRIANGULATED THIS IS PROGRAM ERROR.
   PLEASE
   CONTACT MSC.
   USING PARAM,DUMSEGS,ON CAN SOLVE THIS TRIANGULATION PROBLEM.
   ```

   ```
   %E-P4309907-V4_SUBELEM_CONNECT_EXTENDED_GPS, ,
   DUMMY SURFACE CANNOT BE TRIANGULATED THIS IS PROGRAM ERROR.
   PLEASE
   CONTACT MSC.
   USING PARAM,DUMSEGS,ON CAN SOLVE THIS TRIANGULATION PROBLEM.
   ```

   In that case PARAM,DUMSEGS,ON can maintain a good dummy surface.

   The quality of the dummy surface can be checked by requesting a coupling surface archive. This is done by using the CPLSURFS command and by putting option DUMMY of PARAM, AUTOCOUP to ON.

   For all holes that are not internal holes, the standard method is used.

ELDLTH

Show List of Lagrangian Elements with Time Step in Ascending Order

When you include this parameter in the input file, a list of "n" elements with the time step in ascending order is listed to the output file at the end of time step zero.

Format and Example	Default
`PARAM,ELDLTH,value`	No list printed.
`PARAM,ELDLTH,100`	

Option	Meaning	Type
`value`	Number of elements to be listed	I > 0

Remarks

1. When the value is set to zero, a list of all Lagrangian (structural) elements is printed to the output file.
2. Spring and damper elements are not shown in the list.
3. When you use this feature to determine the initial time step, you should be aware that the time step listed includes the time-step safety factor.

ERRUSR Redefinition of Severity and Number of Error Message Prints

Redefinition of severity and number of prints of error messages.

Format and Example
`PARAM,ERRUSR,name,severity,prints`
`PARAM,ERRUSR,P2010053,I,2`

Option	Meaning		Type	Default
`name`	Error name		C	Required
`severity`	Error severity:		C	Required
	I	Informative		
	W	Warning		
	E	Error		
	F	Fatal		
	C	Catastrophic		
	N	Nastran–ignore		
`prints`	Number of times the message is printed		I	5

Remarks

1. An error name consists of a maximum of eight characters. The entry is also used as a wildcard by entering less than eight characters. The string then is matched with the actual names, and every match redefines the actual message.
2. See Chapter 8: Diagnostic Messages in this manual.

EULER-BOUNDARY

Euler Boundary Treatment

Defines boundary treatment for Euler boundaries.

Format and Example	Default
`PARAM,EULER-BOUNDARY, option`	`ELEMENTVALUE`

Option	Meaning	Type
`EXTRAPOLATION`	The pressure that a wall or coupling surface exerts on the adjacent Euler element is obtained from extrapolating the element pressure toward this boundary.	C
`ELEMENTVALUE`	The pressure that a wall or coupling surface exerts on the adjacent Euler element equals the pressure inside this element.	C

Remarks

1. The finite volume representation, in general, assumes that element values are constant within each element. While this assumption is adequate for the large majority of applications, fluid models involving hydrostatic pressure gradients require that the pressure gradient also be recognized to exist within the element. When element-internal hydrostatic gradients are not accounted for, the calculation is less accurate and suffers from numerical symptoms like pair forming of element pressures. By activating the `EXTRAPOLATION` option, hydrostatic gradients inside the element are taken into account. For meshes without bias, `EXTRAPOLATION` option only modifies the numerical schemes along the boundary.
2. When coupling surfaces are used, `PARAM, FASTCOUP` has to also be activated.

EULERCUB

Cube Sorting for Euler Domains

Divides an Euler domain into several cubes.

Format and Example	Default
PARAM,EULERCUB,NELCUBE,NBX,NBY,NBZ	
PARAM,EULERCUB,2000,2,2,2	

Option	Meaning	Type
NELCUBE	The number of elements per cube. This number is used as a guideline. The actual number used per cube can differ and can be found in the OUT file.	See Remark 1.
NBX	Overrules NELCUBE. The number of cubes in the x-direction.	See Remark 2.
NBY	The number of cubes in the y-direction.	
NBZ	The number of cubes in the z-direction.	

Remarks

1. By setting NELCUBE equal to 2000, optimal use is made of memory caching during Euler computation. This can give a speedup of 1.5. When using adaptive meshing with DMP, additional Euler cubes are created during the simulation. To keep the total number of cubes that are created within bounds, the initial number of cubes should be limited to 100.
2. If NBX is defined, also NBY and NBZ need to be defined. Defining NBX overrules the definition of NELCUBE.
3. There are several ways to distribute cubes across processors. Some ways may lead to bad load balancing. To avoid this it is possible to control the way cubes are distributed across processors by defining PARAM,EULERPR. To check the load balancing the number of active Euler elements on each processor is written out every 1000 cycles.
4. Only supports Euler domains created by **MESH**,BOX. Limitations are:
   - No CHEXA's.
   - No FLOW or WALLET. Boundary conditions have to be defined by FLOWDIR, FLOWTSQ, or WALLDIR.
   - Time history output of adaptive Euler elements is not supported.
5. When several cubes are on each processor, it may be useful to switch on option 5 of **PARAM,FASTCOUP** with VALUE=ALL.

EULERPR

Euler Cube Partition Control

Controls the partitions the Euler cubes on the processors.

Format and Example
PARAM,EULERPR,PROCDIR,NPX,NPY,NPZ
PARAM,EULERPR,USER,2,2,2

Option	Meaning		Type	Default
PROCDIR	This directive controls the way cubes are distributed across processors. The effect can be checked by checking the Eulerian output variable PARTITION.		C	X
	X	Partition in global X direction first.		
	Y	Partition in global Y direction first.		
	Z	Partition in global Z direction first.		
	USER	Define user defined partitioning.		
	SIMPLE	Partition Euler cubes in a simple pattern.		
	ORB	Partitions using orthogonal recursive bisection. See remark 5.		
NPX	The number of partitions in the x-direction. Required for PROCDIR=USER.		I	0
NPY	The number of partitions in the y-direction.		I	NPX
NPZ	The number of partitions in the z-direction.		I	NPX

Remarks

1. There are several ways to distribute cubes across processors. Some ways may lead to bad load balancing. To avoid this it is possible to control the way Euler cubes are distributed across processors by defining PROCDIR.
2. When option PROCDIR=USER, the values for NBX, NBY, and NBZ must be such that NBX is equal or a multiple of NPX, NBY is equal or a multiple of NPY and NBZ is equal or a multiple of NPZ. Also for this option, NPX*NPY*NPZ must be equal to the number of processors used in the cluster.
3. For option PROCDIR=SIMPLE, the values NBX, NBY, and NBZ on PARAM , EULERCUB must be such that NBX*NBY*NBZ is equal or a multiple of the number of processors used. For instance, if the number of processors in the cluster is 4, NBX*NBY*NBZ must be equal to either 4 or 8 or 12, etc. Otherwise, the calculation will terminate prematurely with an error message.

4. Only supports Euler domains created by **MESH**, BOX and **MESH,**ADAPT. Limitations are:
   - No **PEULERx/CHEXA's**.
   - No FLOW or WALLET entries are allowed. Boundary conditions have to be defined by FLOWDIR, FLOWTSQ, or WALLDIR.
5. With the orthogonal recursive bisection method, the mesh is repeatedly cut in half. Each cut is along a coordinate direction, and the direction of the cut is chosen to minimize the communication cost. To illustrate this method, consider, for example, a mesh with 200, 150, and 50 elements in, respectively, the x- ,y-, and z-direction. When split in x-direction, the number of Euler faces at the split will be 150*50 = 7500. When splitting across the y-direction, it will be 200*50=10000, and for the z-direction, it will be 200*150=30000. Since the x-split has the smallest number of faces at the split, the x-split has the smallest communication cost. Therefore, the ORB scheme will select the x-split. At the x-split, there are two mesh parts. Both have 100*150*50 elements. Now a split in y-direction will give the minimal communication costs. This process of bisecting is continued until the number of sub meshes equals the number of CPU's.

EULSTRESS

Stress Update Method

Defines the update logic for stresses when material is transported in Euler elements.

Format and Example	Default
PARAM,EULSTRESS,option	VOLUME
PARAM,EULSTRESS,MASS	

Option	Meaning	Type
MASS	Update stresses by transporting mass*stress.	C
VOLUME	Update stresses by transporting volume*stress.	

Remarks

1. Only used for the MMSTREN solver.
2. Stresses are a material property and when material flows in or out an element the stress state in the element is changed. This is analogous to temperature and energy. Not the temperature is transported, but energy. After transporting energy the temperature is recomputed by dividing the energy by element mass and specific heat. In case of stress, the "energy" is given by mass times stress. After transporting this "energy" the new stress follows by dividing it by mass. As shown in *Dytran Theory Manual*, Chapter 6: The Standard Euler Solver, Euler With Strength, this gives a correct updating procedure for stresses. There it also proven that stress times mass is conserved during transport.
3. In most simulations, variations in density are small and one can replace multiplication by mass by a multiplication by volume. This method is activated by the VOLUME option which is also the default option. Using the MASS option may have some influence on simulations with large density variations. The MASS option gives the most accurate results.
4. The transport logic of the effective plastic strain is identical to that of stresses. When using the MASS option, the plastic strain is computed more accurately when material is compressed.
5. The Euler with Strength solver always uses the MASS option, by default.

EULTRAN Switch for the Euler Transport Scheme of the Multi-material Solver and the Single Material Strength Solver

Sets the definition of the face velocity used in the transport scheme of the Multi-material solver and the single material strength solver.

Format and Example	Default
PARAM,EULTRAN,option	IMPULSE
PARAM,EULTRAN,AVERAGE,FAIL	NOFAIL

Option	Meaning	Type
IMPULSE	The face velocity is impulse weighted	C
AVERAGE	The face velocity is a simple average	C
FAIL	Failure is transported. See Remarks 5.and 6.	C

Remarks

1. The default value of IMPULSE is sufficient for most Euler problems. Especially problems where the reference density of the different materials varies widely (for example, orders of magnitude) are required to use the default option.
2. In case the IMPULSE option (default) is used, the Euler transport scheme computes that the face velocity uses an impulse weighted average of the material velocity in the left and the right element adjacent to the face.
3. In case the AVERAGE option is used, the Euler transport scheme computes the face velocity as one-half times the sum of the material velocity in the left and the right element adjacent to the face.
4. Does not apply to the single material hydrodynamic solver and the Roe solver.
5. The FAIL option requires a failure model for at least one Eulerian material. In case of the default NOFAIL, then failed Euler material can support shear stress again as soon as new material enters the Euler element. Thus the information that part of the material inside the Euler element has failed is lost. The FAIL option activates transport of fail fraction and thereby keeps track of material that has failed. In this way only the failed part of the element can no longer supports shear stresses. In more detail, the yield stress in the material is scaled by (1-failfrac), where failfrac denotes the fail fraction of the material. The fail fraction of the first material in an element can be retrieved from Euler archive or time-history results files in the DAMAGE variable. The value of fail fraction DAMAGE is between zero and one.
6. The FAIL option cannot be combined with the Johnson-Cook failure model (FAILJC).

EUSUBCYC — Growth or Subcycling Interval in Euler Computations

Controls the growth of the subcycling interval in Euler computations.

Format and Example	Default
PARAM,EUSUBCYC,value	1

Option	Meaning	Type
value	Maximum growth of the subcycling interval.	1 > 0

Remarks

1. The subcycling algorithm automatically estimates the number of subcycles to be used. This is updated throughout the calculation. This parameter controls how much the number of subcycles can grow. For example, **EUSUBCYC** is set to 1, and the current number of time steps between updates of the Euler variables. If Dytran estimates that the subcycling interval should be 7, the subcycling interval is increased by 1 until a value of 7 is reached.
2. There is no control on the amount by which the subcycling interval can decrease.

EUSUBMAX

Subcycle Limit in Euler Solver

Defines the maximum number of subcycles that can occur in the Euler solver. During a subcycle, the Euler computations are skipped.

If coupling surface computations are more expensive than Euler computations, then use of COSUBMAX should be considered first. In that case, the optimal setting of **EUSBMAX** is `LINKCS = BOTH` and `FVUMAX` is blank.

As with COSUBMAX, use of **EUSUBMAX** can lead to loss of accuracy for certain simulations. It is recommended to validate the use of **EUSUBMAX** by comparing the difference in results between using **EUSUBMAX** and not using **EUSUBMAX** for some typical target simulations.

Format and Example	Default
PARAM,EUSUBMAX,NSUBMAX,DFVUMAX,LINKCS	For LINKCS: BOTH

Option	Meaning	Type
`NSUBMAX`	The maximum number of time steps between updating Euler variables.	1 > 0 See Remark 1.
`DFVUMAX`	Maximum allowed increase in uncovered volume fraction between two subsequent Eulerian computations.	See Remark 5.
`LINKCS`	Specifies interactions between Euler subcycling and coupling surface subcycling.	See Remark 6.

Remarks

1. Updating the coupling geometry can take a lot of CPU time. Subcycling can give substantial savings in CPU time for coupled calculations.
2. The smaller the value of `DFVUMAX`, the greater the accuracy of the analysis and the greater the cost. Conversely, larger values offer significant CPU savings, but very large values give incorrect results.
3. When the Euler time step is considerably larger than the Lagrange time step, it can be worthwhile to skip the Euler computation for several cycles. This can reduce computational costs considerable.
4. Skipping the Euler computation for several cycles leads to postponed time steps. When the Euler computations are done, these have to be taken into account by fluxing with an accumulated time step. Skipping the Euler computation is stopped as soon as the accumulated time step becomes larger than the stable Euler time step. The total number of skipped cycles is limited by `NSUBMAX`. `NSUBMAX` is required input.
5. In skipping the Euler computation, the movement of the coupling surface has to also be monitored. To estimate this movement, the change in uncovered fraction of the elements is used. If the change in uncovered, volume fraction from one cycle to the other is larger than `DFVUMAX`; then the Euler computation is not skipped that cycle. `DFVUMAX` has to be left blank if coupling surface subcycling is used. Coupling surface subcycling already monitors the movement of the coupling surface. Choosing too large values for `DFVUMAX` can make results inaccurate.

6. **EUSUBMAX** can be used with COSUBMAX. LINKCS specifies how the two subcycling processes influence each other. Allowed values for LINKCS are:

BOTH	The Euler computations will not be skipped when the coupling surface computations have been done. On the other hand, if an Euler computation is to be done, a coupling surface computation is done also. The number of times that the Euler computation is skipped equals the number of times that the coupling surface computation has been done.
COUPLE	If an Euler computation is to be done, also a coupling surface computation is done.
EULER	The Euler computations will not be skipped when the coupling surface computations have been done.
NONE	The Euler subcycling and coupling surface subcycling are independent. DFVUMAX has to be defined

FAILDT

Element Time-step Based Failure Model

Defines the property of a failure model where element failure occurs when the element's time step falls below the specified limit.

Format and Example	Default
`PARAM,FAILDT,value`	1.E-20

Option	Meaning	Type
`value`	Minimum time-step	R > 0.0

Remarks

1. This failure model is valid for all Lagrangian solid (CHEXA)and shell (CQUAD4)elements.
2. All elements for which the time step falls below the specified value are removed from the computation.
3. Although it is not usually necessary to limit the element time-step for Lagrangian elements, there are occasions where specifying a minimum time-step can be advantageous for computational performance, for example, when adaptive contact is used.
4. Note that this parameter should be used with care as you may influence the results of the analysis when you set the time-step criterion to a too high value. You then run the risk that elements are removed from the analysis while they may still be relevant.).

FAILOUT

Failed Element Output Parameter

Defines whether failed elements are written to the output file (ARCHIVES).

Format and Example	Default
PARAM,FAILOUT,option	YES
PARAM,FAILOUT,NO	

Option	Meaning	Type
NO	Failed elements are not written to the archive files.	C
YES	Failed elements are written to the archive files.	C

Remarks

1. When the NO option is chosen, the archives are written out as one file per requested time step regardless of the number set in the SAVE command for the archive files that appear in the Case Control Section.
2. Failed elements are NOT filtered when written to a RESTART file or a TIMEHISTORY file.f.

FALENEG Eroding energy control parameter

Defines whether eroding energies are separately stored in OUT and THS file.

Format and Example	Default
PARAM,FALENEG,option	OFF
PARAM,FALENEG,ON	

Option	Meaning	Type
OFF	Eroding energies are not calculated and not stored separately.	C
ON	Eroding energies are calculated and stored separately.	C

Remarks

1. When it is set to “ON”, MATOUT output request can store EKINF (eroding kinetic energy), EINTF (eroding internal energy) and EHRGF (eroding hourglass energy) information in THS file.
2. When it is set to “ON”, EROENGY-KN (eroding kinetic energy), EROENGY-IN(eroding internal energy) and EROENGY-HG (eroding hourglass energy) will be written in material summary of OUT file.
3. When it is set to “ON”, the total energy will be sum of kinetic energy, internal energy, hourglass energy, eroding kinetic energy, eroding internal energy and eroding hourglass energy.
4. When it is set to “ON”, EDIS (distortional energy) and SIE (specific internal energy) of ELOUT request will be set to zero.
5. This parameter is only available for Lagrangian elements.

FASTCOUP

Fast Coupling Algorithm

Defines the fast coupling algorithm.

Format and Example	Default
`PARAM,FASTCOUP,option1,option2,option3,option4,option5,option6`	See Remark 1.
`PARAM,FASTCOUP,INPLANE,FAIL,NOCHKEUL,NEARONLY,NONE,OFF`	

Option	Meaning		Type
`option1`	`blank`	The default is used.	See Remark 2.
	`INPLANE`	Small offset for inplane coupling surface segments.	
	`NO-OFFSET`	No offset for inplane coupling surface segments.	
`option2`	`NOFAIL`	No failure of the coupling surface.	See Remark 3.
	`FAIL`	Failure of the coupling surface will be taken into account.	
`option3`	`CHKEUL`	Checking whether all segments of the coupling surface are fully inside Euler elements.	
	`NOCHKEUL`	No checking.	
`option4`	NEARONLY	Do the full coupling surface only in the first cycle. In subsequent cycles only update elements and faces that are near the coupling surface.	`ALL`
	ALL	Do the full coupling surface computation each cycle.	
`option5`	Controls coupling surface computations when there are multiple cubes.		
	NONE	All coupling surface computations are done for each cube.	`NONE`
	FACES	The face cover fraction computation is done in one go for all cubes. The Other coupling computations are still done per cube.	
	POLPK	The polpack computation efficiently handles multiple cubes. The other coupling computations are still done per cube.	
	ALL	All coupling surface computations efficiently handle cubes.	

Option	Meaning		Type
option6	In the coupling surface computations, it is determined which structural segments intersect the Euler mesh. This is done by looping across all structural segments. For dmp, each CPU has only part of the Euler mesh. Therefore, for many segments, it is known beforehand that they will not intersect the Euler mesh of the CPU and can be skipped at an early stage of the computation. Option 6 activates this skipping. (Character; Default = OFF). Option 6 is obsolete and has been replaced by the more efficient option 7.		
	OFF	For dmp, each CPU goes over all segments.	
	ON	each CPU skips, if possible, segments that reside on other CPUs.	
option7	Each coupling surface segment has to be intersected with the Euler elements and Euler faces. For large Euler meshes, a substantial amount of checking has to be done do find the intersecting elements and faces. By dividing the Euler mesh in search boxes, the costs of this checking is kept to a minimum. Especially for large Euler meshes combined with large coupling surface, this checking can become expensive, and then the use of search boxes can significantly reduce the costs. For Euler meshes of less than 100000 elements, the use of search will boxes have little effect. (Character; Default = ON)		
	OFF	For dmp, each CPU goes over all segments.	
	ON	Each CPU skips, if possible, segments that reside on other CPUs.	

Remarks

1. Default value for option1 is INPLANE and for option2 NOFAIL.
2. When option1 is set to INPLANE or when option 1 is blank, a small offset is given to coupling surface segments that are on top of a face of an Eulerian element. This is done because coupling surfaces segments on Eulerfaces make the Euler element volume computation invalid. Also boundary conditions on these segments are not correctly imposed. The net effect of these problems is unpredictable. The problem can either run correctly, or remain stable but give false results or become instable. The option NO-OFFSET is obsolete and should not be used.
3. Option2 can only be used in combination with PARAM, LIMITER, ROE, or MMHYDRO or MMSTREN. The coupling surface must consist of CQUADs and/or CTRIAs and a failure model for the material of the surface must be defined.
4. This parameter can only be used when the Eulerian mesh is aligned with the basic coordinate system axes.
5. When there are four or more Euler cubes per processor, option 5 is set to ALL. For less Euler cubes ,option 5 will not give significant speed-up.

6. To get optimal speed-up for dmp or when using multiple Euler cubes, it is not necessary to change options. The options 5, 6, and 7 can be left blank as the default values are already optimal.

FBLEND

Blend Fraction

Eulerian elements with uncovered fractions smaller than FBLEND are blended with adjacent elements to form a clump so that they do not control the time step.

Format and Example	Default
PARAM,FBLEND,value	0.6667
PARAM,FBLEND,0.5	

Option	Meaning	Type
value	The uncovered fraction below which blending occurs.	0.0 ≤ R < 1.0

Remarks

1. The default value is satisfactory for virtually all calculations.
2. Elements are blended only if they would have controlled the time step otherwise.
3. Elements with uncovered fractions greater than FBLEND are not blended and are allowed to control the time step.
4. Large values of FBLEND produce a larger time step but many blends. Small values produce a smaller time step and fewer blends.
5. In a calculation with a coupling surface, STEPFCT is smaller or equal FBLEND to avoid instabilities (see PARAM,STEPFCT).

FLOW-METHOD

Flow-Method Between Two Euler Domains Across Open Areas in Coupling Surfaces

Defines the method for simulating material flow between two Euler domains across open areas in coupling surfaces.

Format and Example	Default
`PARAMS,FLOW-METHOD,option`	POLPACK
`PARAM,FLOW-METHOD,POLPACK`	

Option		Meaning	Type
`option`	`POLPACK`	The facets in the coupling surfaces that represent an open area are subdivided into smaller facets, with each connecting exactly to one Euler element in the first Euler domain and to exactly one Euler element in the second Euler domain. Material flow takes place across these smaller, subdivided facets (POLPACKs). This is the most accurate method.	C
		See Remark 1 for more details.	
	`FACET`	The facets in the coupling surfaces that represent an open area are not subdivided. Material flow takes place across the original facets. If these facets are too large in relation with the Euler elements, the method becomes inaccurate. Material flow across one facet can involve several Euler elements on both sides of the hole and averaging occurs.	

Remarks

1. For a detailed description of the theory involved, see Reference [18.].
2. This parameter applies to simulations where:
   - two coupling surfaces share a common set of facets.
   - each coupling surface has it's own Euler domain.
   - material flows from one Euler domain into the other through the open area represented by the common set of facets.

   Flow only occurs if:
   - the common facets are defined as 'open', using `PORFLCP` or `PORFCPL`.
   - the common facets open up due to failure of a shell structure, using `COUP1INT`.

Examples simulations are:

- Holes between air bag compartments.
- Holes between containers filled with gas or liquid.
- Open area between the top of a fuel-tank baffle and the fuel-tank skin.
- Open area in-between wide straps inside an air bag.
- Failure of walls in between aircraft wing compartments.
- Failure of tank armor by a blast wave.
- Etc.

3. The following table summarizes what input cards support the simulation of material flow between two Euler domains across an open areas in coupling surfaces:

Table 1:

Euler Solver	Material flow through a coupling surface	FLOW-METHOD = POLPACK	FLOW-METHOD = FACET	PORFLCPL (velocity based	PORFCPL (Pressure based)	COUP1INT/COUP1FL (Failure of shell elements creates the opening)
HYDRO	YES	YES	NO	YES	YES	Only for flow-method = polpack
HYDRO – Roe solver – 1st Order	YES	YES	YES	YES	YES	YES
HYDRO – Roe solver – 2nd Order	YES	YES	YES	YES	YES	YES
MMHYDRO	YES	YES	NO	YES	NO	Only for flow-method = polpack
STRENGTH	NO	-	-	-	-	-
MMSTREN	YES	YES	NO	YES	NO	Only for flow-method = polpack

4. The Euler domains are shown below with the support types for each:

Table 3-1

	FLOW-METHOD = POLACK	FLOW METHOD = FACET
MESH → TYPE=ADAPT	Yes	No
MESH → TYPE=BOX	Yes	Yes
Modeling of CHEXA elements	No	Yes

5. An Euler domain is associated with a coupling surface by specifying the MESHID or SET1ID on the COUPLE option.
6. FLOW-METHOD = POLPACK has the following limitations:
   - The entries NSTGP and NSTEL on all MESH entries should be left blank. It is not allowed to specify for any MESH entry the starting element number or starting grid point number.
   - There are restrictions on output requests. See the section "Output" of GettingStarted-Multiple-Coupling-Surfaces.
   - Flow faces and wallets are not supported.

   Note: flowdef is supported.

   - Viscosity is not supported

   A case where these limitations require the use of FLOW-METHOD = FACET is when the Euler elements are generated in Patran, not using the MESH option, and one or more of the following options is used:
   - FLOW boundaries are defined on some Euler faces.
   - WALLET boundaries are defined on some Euler faces.
   - Viscosity is defined.

FMULTI — Multimaterial Overflow Array Parameter

Defines the dimension of the multimaterial element array.

Format and Example	Default
`PARAMS,FMULTI,value`	.10
`PARAMS,FMULTI,.25`	

Option	Meaning	Type
`value`	The relative amount of multimaterial elements.	0 < R < 1.

Remark

The multimaterial Eulerian elements use an overflow array in which to store material data. This array can hold `FMULTI` times the total number of Eulerian elements. In a problem where more than 10% of the elements have more than one material, the default value of `FMULTI` must be increased.

GEOCHECK

Define Geometry Consistency Check

This parameter forces a check of the geometry for consistent connectivity of the defined hexagonal elements and correction if needed.

Format and Example	Default
`PARAM,GEOCHECK,option`	OFF
`PARAM,GEOCHECK,ON`	

Option	Meaning	Type
`ON`	Geometry consistency check is performed.	C
`OFF`	No geometry consistency check is performed.	

Remarks

1. The defined geometry is checked for consistent connectivity of the hexagonal elements. If an inconsistency is detected, the connectivity is corrected. CFACE entries with references to elements that have been corrected are corrected as well.
2. If a hexagonal mesh is generated with other commercial preprocessors, this parameter can correct the connectivity of the hexagonal elements in case problems are encountered with face generation or volume computation.

GRADED-MESH

Glue Sets of Euler Elements

Glues fine meshes to coarse meshes. See the section on Graded meshes in the user manual for further information.

Format and Example	Default
`PARAM,GRADED-MESH,option`	`MINVOLUME`
`PARAM,GRADED-MESH,MINVOLUME`	

Option	Meaning		Type
`Option`	`MINVOLUME`	If an element of one mesh is covered by an element of another mesh the element with the largest volume will be inactivated. It will also be removed from the output request for Eulerian archives.	C
	`ELNUM`	If an element of one mesh is covered by an element of another mesh the element with the smallest element number will be inactivated. It will also be removed from the output request for Eulerian archives.	

Remarks

1. This parameter can be used to build block-structured meshes.
2. All Euler elements have to be either orthogonal or axial symmetric.
3. To get meaningful physical results, the change in mesh size going from one element to the next should not be larger than 1.4 or smaller than 0.7.

HGCMEM Shell Membrane Hourglass Damping Coefficient Parameters

Defines the default membrane damping coefficient for shell elements.

Format and Example	Default
`PARAM,HGCMEM,value`	See Remark 3.
`PARAM,HGCMEM,0.07`	

Option	Meaning	Type
`value`	Hourglass damping coefficient	$0.0 \leq R \leq 0.15$

Remarks

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.
3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for shell elements is either equal to the default value of 0.1 or is equal to the default value defined on a `PARAM,HGCOEFF` entry.
4. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HGCOEFF

Hourglass Damping Coefficient

Defines the global default hourglass damping coefficient.

Format and Example	Default
`PARAM,HGCOEFF,value`	See Remark 3.
`PARAM,HGCOEFF,0.14`	

Option	Meaning	Type
`value`	Hourglass damping coefficient	$0.0 \leq R \leq 0.15$

Remarks

1. The default applies to all types of hourglass suppression methods and should be used unless there is good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficients are not explicitly defined on HGSUPPR entries or on a HGCMEM HGCWRP, HGCTWS, or HGCSOL entry.
3. If this entry is omitted, the default value of the hourglass damping coefficients is either equal to the default value of 0.1 or is equal to the value specified on a HGCMEM, HGCTWS, HGCWRP, or HGCSOL PARAM entry.
4. The value of the coefficients can be explicitly defined for each property by using an HGSUPPR entry.
5. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HGCSOL

Solid Hourglass Damping Coefficient

Define the default damping coefficient for solid elements.

Format and Example	Default
`PARAM,HGCSOL,value`	See Remark 3.
`PARAM,HGCSOL,0.11`	

Option	Meaning	Type
`value`	Hourglass damping coefficient	$0.0 \leq R \leq 0.15$

Remarks

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.
3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for solid elements is either equal to the default value of 0.1 or is equal to the default value defined on a `PARAM,HGCOEFF` entry.
4. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HGCTWS — Shell Twisting Hourglass Damping Coefficient

Defines the default twisting damping coefficient for shell elements.

Format and Example	Default
`PARAM,HGCTWS,value`	See Remark 3.
`PARAM,HGCTWS,0.02`	

Option	Meaning	Type
`value`	Hourglass damping coefficient	$0.0 \leq R \leq 0.15$

Remarks

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.
3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for shell elements is either equal to the default value of 0.1 or is equal to the default value defined on a `PARAM,HGCOEFF` entry.
4. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HGCWRP — Shell Warping Hourglass Damping Coefficient

Defines the default warping damping coefficient for shell elements

Format and Example	Default
`PARAM,HGCWRP,value`	See Remark 3.
`PARAM,HGCWRP,0.0`	

Option	Meaning	Type
`value`	Hourglass damping coefficient	$0.0 \leq R \leq 0.15$

Remarks

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.
3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for shell elements is either equal to the default value of 0.1 or is equal to the default value defined on a `PARAM,HGCOEFF` entry.
4. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HGSHELL — Shell Hourglass Suppression Method

Defines the default hourglass suppression method for shell elements

Format and Example	Default
`PARAM,HGSHELL,option`	See Remark 2.
`PARAM,HGSHELL,DYNA`	

Option	Meaning	Type
`FBV`	Flanagan-Belytschko stiffness hourglass damping	C
`DYNA`	Viscous hourglass damping	

Remarks

1. The type of hourglass suppression method defined on this entry is used as the default whenever the type fields on the HGSUPPR for shell properties are left blank.
2. If this entry is omitted, the default suppression method used for shell elements is either `FBV` or the default method defined on the `PARAM,HGTYPE` entry.
3. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HGSOLID

Solid Hourglass Suppression Method

Defines the default hourglass suppression method for solid elements.

Format and Example	Default
`PARAM,HGSOLID,option`	See Remark 2.
`PARAM,HGSOLID,FBS`	

Option	Meaning	Type
`FBS`	Flanagan-Belytschko stiffness hourglass damping	C
`DYNA`	Viscous hourglass damping	

Remarks

1. The type of hourglass suppression method defined on this entry is used as the default whenever the type fields on the HGSUPPR for solid properties are left blank.
2. If this entry is omitted, the default suppression method used for solid elements is either FBS or the default method defined on the `PARAM,HGTYPE` entry.
3. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HGTYPE

Hourglass Suppression Method

Defines the default type of hourglass suppression method.

Format and Example	Default
`PARAM,HGTYPE,option`	See Remark 2.
`PARAM,HGTYPE,FBS`	

Option	Meaning	Type
`FBS`	Flanagan-Belytschko stiffness hourglass damping for solids	C
`FBV`	Flanagan-Belytschko stiffness hourglass damping for shells	
`DYNA`	Viscous hourglass damping for shells	

Remarks

1. The type of the hourglass suppression method defined on this entry is used as the default whenever the type fields in the HGSUPPR entries are left blank.
2. If this entry is omitted, the type can be defined on a `PARAM,HGSHELL` entry for shell elements, a `PARAM,HGSOLID` entry for solid elements, or on the HGSUPPR entries; otherwise the defaults apply. For shell elements the default is `FBV`; for solid elements, the default is `FBS`.
3. See *Dytran Theory Manual*, Chapter 4: Models, Artificial Viscosities for details on hourglass suppression methods.

HICGRAV

Gravity Used by HIC Calculations

Defines the value of the gravity to be used by the HIC calculations

Format and Example	Default
`PARAM,HICGRAV,value`	9.80665
`PARAM,HICGRAV,980.7`	

Option	Meaning	Type
`value`	Gravity used by HIC Calculations	R > 0.0

Remark

1. The value set by this parameter will be used by all HIC output requests.
2. This parameter can only be set once in the input deck.

HVLFAIL

Failure at Hydrodynamic Volume Limit

Defines element failure on the hydrodynamic volume limit.

Format and Example	Default
PARAM,HVLFAIL,option	NO
PARAM,HVLFAIL,YES	

Option	Meaning	Type
YES	Element failure on hydrodynamic volume limit	C
NO	No element failure on hydrodynamic volume limit	

Remarks

1. Lagrangian elements (CHEXA) that have a material model with a failure model fail when the hydrodynamic volume limit is reached and the parameter is set to YES.

   The elements can fail only when the following items are presented in the input:

   a. The material model has a hydrodynamic volume limit (HVL).

   b. A failure model is defined.

   c. PARAM, HVLFAIL, YES

   Example

   ```
   PARAM, HVLFAIL, YES
   DMAT, 1, 7850, 101, 102, 103, 104
   EOSPOL, 101, 175.E9,,,,,,+
   +, 1.1
   SHREL, 102, 80.77E9
   YLDVM, 103, 1.E20
   FAILMPS, 104, 1.E20
   ```

2. The hydrodynamic volume limit by default allows for 10% expansion.

HYDROBOD

Hydrodynamic Body Force

Defines a body force for single hydrodynamic material in Euler.

Format and Example	Default
`PARAM,HYDROBOD,XACC,YACC,ZACC`	No body force applied
`PARAM,HYDROBOD,-300.,0.,150.`	

Option	Meaning	Type
`XACC`	X-acceleration.	R
`YACC`	Y-acceleration.	R
`ZACC`	Z-acceleration.	R

Remark

This parameter defines a constant body force load in Euler for single hydro material only.

IEEE

IEEE Binary Data Output Format

On CRAY computers, archive and time-history output is defined in IEEE format rather than in Block Data format

Format and Example	Default
`PARAM,IEEE,option`	`OFF`
`PARAM,IEEE,ON`	

Option	Meaning	Type
`ON`	Activate IEEE output	C
`OFF`	No IEEE output	

Remarks

1. On computers that, by default, write binary data in IEEE format, the parameter will have no effect.
2. Binary IEEE files can be transported to all IEEE computer systems. (Note that this in only valid for `ARCHIVE` and `TIMEHISTORY` files.)

IGNFRCER

Ignores Warnings

Ignores certain errors for FORCE and MOMENT entries.

Format and Example	Default
PARAM,IGNFRCER	See below

Option	Meaning
No option	Some warnings that are given when using the FORCE1 FORCE2, MOMENT1, or MOMENT2 entries are normally accompanied by an additional error message. By using this PARAM, the warnings are still issued, but the error message is not issued.

IMM

Initial Metric Method Formulation

The option allows to specify the IMM method to be used.

Format and Example	Default
PARAM,IMM,option1,option2,STREC,DTREC	FULL (Option 1)
PARAM,IMM,ZERO,YES,1.0E-3,1.0E-3	

Option	Meaning		Type
Option1	FULL	While elements are under IMM condition, they will carry stresses when under compression.	C
	REDUCED	While elements are under IMM condition they will carry a reduced stress when under compression. The relative area factor SMDFER is used to reduce the Young's modulus.	
	ZERO	While elements are under IMM condition they do not carry any compressive stresses. Use material damping to avoid excessive nodal velocities.	
Option2	NO or OFF	Do not recalculate IMM strains during the calculation. (See Remark 3.)	C
	YES or ON	Recalculate IMM strains during the calculation. (See Remark 3.)	
STREC	Start time of recalculation of IMM strains. (Default is 1.0E-3; see Remark 4.)		R > 0.0
DTREC	Times between recalculation of IMM strains. (Default is 1.0E-3; See Remark 4.)		R > 0.0

Remarks

1. Method ZERO is best suitable when initially more than a couple of elements with zero or near zero area are present in the model.
2. The Initial Metric Method is described in the *Dytran User's Guide* in Chapter 6: The Standard Euler Solver, Initial Metric Method for Air Bags.
3. The default for Option2 depends on Option1.

Table 4:

Option1	Default for Option2
FULL	OFF
REDUCED	OFF
ZERO	ON

4. When Option2 is `OFF` or `NO`, `STREC` and `DTREC` are neglected.

INFO-BJOIN

List the Generated BJOINs and Spotwelds

Additional information about the BJOIN and spotweld connectivity will be listed in the output file. The information listed is:

- Grid point pairs forming a BJOIN or a spotweld.
- BJOINs and spotwelds initially connected.

Format and Example	Default
`PARAM,INFO-BJOIN,option`	`NO`
`PARAM,INFO-BJOIN,YES`	

Option	Meaning	Type
`YES`	Information is issued	C
`NO`	Information is not issued	

INISTEP

Initial Time Step

Defines the time step used at the start of the analysis.

Format and Example	Default
`PARAM,INISTEP,value`	No default
`PARAM,INISTEP,1.E-6`	

Option	Meaning	Type
`value`	Time step (in analysis time units) used for the first iteration	R > 0.0

Remarks

1. This parameter is required to start an analysis.
2. See Chapter 9: Running the Analysis, Controlling the Analysis in the *Dytran User's Guide* for details on time step control.

INITFILE

Method of Initialization Definition

Defines the method of initializing a transient analysis from a Dytran solution file.

Format and Example	Default
`PARAM,INITFILE,option`	`V3`
`PARAM,INITFILE,V1`	

Option	Meaning	Type
`V1`	Version of initialization, where the prestress and the transient input files must obey the following rules: ■ The number of structural elements must be the same. ■ The number of structural grid points must be the same. ■ The boundary conditions on the grid points must be the same. ■ The material models must be the same. ■ Eulerian grid points are allowed to be present in the prestress analysis, but are not written to or read from the Solution file.	C
	This version is available for the following element types: ■ One-dimensional elements ■ Shell elements (including composites) ■ Membrane elements ■ Lagrangian solid elements	
`V2`	Version of initialization, where the prestress and the transient input files must obey the following rules: ■ No restrictions on the number of elements and grid points. ■ No restrictions on the consistency of the boundary conditions. ■ Eulerian grid points are allowed to be present in the prestress analysis, but are not written to or read from the Solution file.	C
	This version is available for the following element types: ■ Shell elements (excluding composites) ■ Lagrangian solid elements	

Option	Meaning	Type
V3	Version of initialization, where the prestress and the transient input files must obey the following rules: ■ The number of structural elements must be the same. ■ The number of structural grid points must be the same. ■ The boundary conditions on the grid points are allowed to change. ■ The material models must be the same. ■ When the Eulerian grid points are present in the prestress analysis they are written to the solution file during the prestress analysis and read from the solution file during the transient analysis.	C
	This version is available for the following element types: ■ One-dimensional elements ■ Shell elements (including composites) ■ Membrane elements ■ Lagrangian solid elements ■ Eulerian elements	

Remarks

1. The user is responsible for consistency upon choosing the V2 definition.
2. See Chapter 9: Running the Analysis, Restarts and Prestress Analysis in the *Dytran User's Guide* for more detailed information about prestress analyses.

INITNAS — Defines the Type of Displacement Initialization File

Defines the type of file to be used for initialization from an MSC Nastran prestress analysis.

Format and Example	Default
PARAM,INITNAS,option	XL
PARAM,INITNAS,XL	

Option	Meaning	Type
XL	File is an MSC.XL export file from a MSC Nastran database (*.xdb).	C
PATRAN	File is an Patran displacement output file from NASPAT.	C
PUNCH	File is an MSC Nastran punch file for the displacements.	

Remarks

1. When Dytran uses the results of an MSC Nastran analysis to start a transient analysis from a prestressed state, the grid-point displacement field, as computed by MSC Nastran, is read from a formatted file written either by MSC.XL, Patran, or MSC Nastran.

   The format of the formatted import files is as follows:

   MSC.XL Export File:

Record 1:	Header 1
Record 2:	Header 2
Record 3:	Header 3
Record 4:	Header 4
Record 5:	Header 5
Record 6:	Header 6
Record 7:	Header 7
Record 8:	Header 8
Record 9 to n+8:	Grid point X-Dis Y-Dis Z-Dis(A8, 3A15)

   Patran Nodal Results Data File

Record 1:	TITLE	(80A1)
Record 2:	NNODES, MAXNOD, DEFMAX, NDMAX, NWIDTH	(2I9,E15.6,2I9)
Record 3:	SUBTITLE 1	(80A1)

Record 4:	SUBTITLE 2	(80A1)
Record 5 to n+4:	NODID, (DATA (j), j=1, NWIDTH)	(I8, (5E13.7))

2. The default is overwritten when input is recognized as a different format.

JWLDET

Blast Wave Ignition of One to Another

Specifies whether the blast wave of one explosive can ignite another explosive. Here it is assumed that the explosives are modeled by a combination of EOSJWL and DETSPH entries.

Format and Example	Default
PARAM,JWLDET,OPTION	
PARAM,JWLDET,NOLINK	

Option	Meaning	Type
LINK	Multiple detonations with EOSJWL are LINKED.	
NOLINK	Multiple detonations with EOSJWL are NOT LINKED.	Default
	The detonation wave of one explosive cannot ignite another explosive.	

Remarks

1. Option NOLINK: TDET is set to -1 for elements that have no JWL material. The NOLINK option is only valid with true JWL materials; not valid for the "Static Detonation/Ideal Gas" model. Here, TDET is an Eulerian output variable that specifies the detonation time. Whether an explosive has ignited can be checked by requesting the Eulerian FBURN output variable. FBURN denotes the burn fraction.
2. Setting this parameter as NOLINK will prevent "sympathetic ignition". Each charge will ignite at the specified TEDT in its own DETSPH card.

LAGPR

Lagrangian Partitioing Control

Controls the partitions of the Lagrangian elements on the processors.

Format and Example
`PARAM,LAGPR, IOPT,xvec,yvec,zvec,xcen,ycen,zcen`
`PARAM,LAGPR, 6, 1.0, 0.0, 0.0, 0.0, -100.0, 0.0`

Option	Meaning	Type	Default
IOPT	Set partitioning type. See Remark 1. 1 Metis best partitioning 2 Metis Decomposition on element based 3 Metis Decomposition on node based 4 Vector Decomposition with Specified User Direction. Xvec, yvec and zvec are required. 5 Radial Decomposition with User Direction. Xvec, yvec, zvec, xcen, ycen and zcen are required. 6 Angular Decomposition with User Direction. Xvec, yvec, zvec, xcen, ycen and zcen are required. 7 Recursive Coordinate Bisection Decomposition	I > 0	1
Xvec	X-component of directional vector.	R	0.0
Yvec	Y-component of directional vector.	R	0.0
Zvec	Z-component of directional vector.	R	0.0
Xcen	X-coordinate of the origin of axis	R	0.0
Ycen	Y-coordinate of the origin of axis	R	0.0
Zcen	Z-coordinate of the origin of axis	R	0.0

Remark

1. To use the option, please check PARAM, DMPOPT.

LIMCUB

Contact Cube Sort Algorithm

Defines the maximum number of cubes used to sort the grid points in a contact definition.

Format and Example	Default
PARAM,LIMCUB,value	2000. See Remark 2.
PARAM,LIMCUB,2300	

Option	Meaning	Type
value	Maximum number of cubes	I > 0

Remarks

1. Each secondary contact node has to search for primary contact nodes that are close enough to have potential contact. It is too expensive to have each secondary contact node check each primary contact node. To limit the number of checks, the space in which the nodes reside is subdivided into cubes. This subdivision is done so that the secondary contact nodes have to check only the primary contact nodes in their own cube and those in the neighboring cubes. The maximum number of cubes used to subdivide the space is equal to the value of LIMCUB.
2. When PARAM,CONTACT,VERSION,SMF is used, the default value of LIMCUB is set to 100,000.

LIMITER

Euler Solver Scheme

Defines the type and the spatial accuracy of scheme used in the Euler solver.

Format and Example	Default
`PARAM,LIMITER,type,option`	See Remark 1.
`PARAM,LIMITER,ROE`	

Option	Meaning		Type
`type`	Type of scheme		C
	`ROE`	Roe solver for single hydro materials	
`option`	blank	Second order in space	See Remark 2.
	`NONE`	First order in space	

Remarks

1. By default, the standard Euler solver is used.
2. By default, second order spatial accuracy is used. The temporal accuracy is defined using the `PARAM,RKSCHEME` entry.
3. When type `ROE` is defined, no void elements are allowed and it cannot be used in combination with EOSJWL. Also, ALE and options concerning air bag analyses are not supported.
4. For more details on the Euler solver see *Dytran Theory Manual*, Chapter 6: Standard Euler Solver.

MATRMERG

Merges MATRIG and RBE2-FULLRIG Assemblies

Parameters

Merges MATRIG and/or RBE2-FULLRIG rigid bodies into a new FULLRIG assembly.

Format and Example	Default
PARAM,MATRMERG,FR<id1>,MR<id2>,MR<id3>,FR<id4>,...	None
PARAM,MATRMERG,FR1,MR2,MR6,MR7,FR4,MR8	
PARAM,MATRMERG,AUTO	

Option	Meaning	Type
FR<id1> or AUTO	Name of the new FULLRIG assembly, or the AUTO option (see Remark 2.)	C
MR<idi> or FR<idi>	Names of MATRIG and/or RBE2-FULLRIG rigid bodies merged into a new FULLRIG assembly with name FR<id1>. No names can be supplied for the AUTO option.	C

Remarks

1. FR<id1> must be a nonexisting RBE2-FULLRIG. The properties of FR<id1> (as mass, center of gravity, and moments of inertia) are computed by Dytran from the properties of each rigid body mentioned on the entry. Rigid body output can be asked for FR<id1>, and loads or rigid body constraints can be applied to FR<id1>. The other MATRIGs and RBE2-FULLRIGs mentioned on the MATRMERG entry disappear after they have been merged.
2. Instead of supplying rigid body names, the AUTO option can be used. After all the normal PARAM,MATRMERG and PARAM,MATRMRG1 entries have been applied, a PARAM,MATRMERG,AUTO merges all the resulting MATRIGs and RBE2-FULLRIGs which have common grid points into a new rigid assembly called FM<id>, where the id is a new FM number starting from 1. As it is not known at the start of an Dytran analysis how many FM-assemblies will be created, no rigid body output can be asked for FM<id>, and no constraints or loads can be applied to FM<id>. The MATRIGs and RBE2-FULLRIGs, which have been merged by the AUTO option into a new FM<id> assembly, disappear.
3. To supply predefined properties for the merged assembly, PARAM,MATRMRG1 can be used, where the first rigid body mentioned on the entry must be an existing RBE2-FULLRIG or MATRIG.

MATRMRG1 Merges MATRIG and RBE2-FULLRIG Assemblies

Merges MATRIG and/or RBE2-FULLRIG rigid bodies into one existing MATRIG or RBE2-FULLRIG assembly with predefined properties.

Format and Example	Default
PARAM,MATRMRG1,MR<id1>,MR<id2>,MR<id3>,FR<id4>,...	None
PARAM,MATRMRG1,MR1,MR2,MR6,MR7,FR4,MR8	

Option	Meaning	Type
MR<id1> or FR<id1>	Name of the new MATRIG or FULLRIG assembly (must be an existing one)	C
MR<idi> or FR<idi>	Names of MATRIG and/or RBE2-FULLRIG rigid bodies, which are merged with the existing MR<id1> or FR<id1> into a new MATRIG assembly, with name MR<id1> or FR<id1>.	C

Remark

MR<id1> or FR<id1> must be an existing MATRIG or RBE2-FULLRIG, respectively. For a FULLRIG, the properties of FR<id1> (as mass, center of gravity and moments of inertia) are computed by Dytran from the properties of each rigid body mentioned on the entry. For a MATRIG, the mass of MR<id1> is either the predefined mass on the MATRIG (id1) entry or the predefined density on the MATRIG (id1) entry times the total volume of all MATRIG members in the MATRMRG1 entry. The center of gravity and moments of inertia of MR<id1> are either predefined on the MATRIG (id1) entry, or are otherwise computed from the properties of each rigid body on the entry. The other MATRIGs and RBE2 FULLRIGs mentioned on the MATRMRG1 entry disappear after they have been merged.

MAXSTEP

Maximum Time Step

Defines the maximum allowable time step.

Format and Example	Default
`PARAM,MAXSTEP,value`	1.E20
`PARAM,MAXSTEP,1.E-3`	

Option	Meaning	Type
`value`	The maximum time step	R > 0.0

Remark

If the time step calculated by Dytran is greater than `MAXSTEP`, the time step is set to `MAXSTEP`.

MESHELL

Mesh Density for Covering Ellipsoids

Defines the factor that determines the mesh density for ellipsoids in contact and for output purposes

Format and Example	Default
`PARAM,MESHELL,value`	5
`PARAM,MESHELL,8`	

Option	Meaning	Type
`value`	Mesh density factor	I > 3

Remarks

1. The mesh density factor is applied for both ellipsoids in a contact definition and for output purposes. For ellipsoids in contact, the default value results in 18 by 36 elements. When you have ellipsoids for output only, the default value results in 8 by 16 elements.
2. The default value is sufficient for most cases. When you increase the value, the representation of the (hyper) ellipsoids is better, but the contact computation will be more expensive, and the archive files will be larger.
3. In case the ellipsoids are meshed for output purposes only, the number of elements in the direction of the ellipsoid's short axis equals $2 \cdot (value\text{-}1)$. The number of elements in the direction of the ellipsoid's long axis is twice the number in the direction of the ellipsoid's short axis.
4. When the ellipsoids are meshed for contact purposes, the number of elements in the direction of the ellipsoid's short axis equals $2 \cdot (2 \cdot value\text{-}1)$. The number of elements in the direction of the ellipsoid's long axis is twice the number in the direction of the ellipsoid's short axis.

MESHPLN

Mesh Density for Covering Rigid Planes

Mesh density for covering rigid planes.

Format and Example	Default
`PARAM,MESHPLN,value`	3
`PARAM,MESHPLN,4`	

Option	Meaning	Type
`value`	Rigid planes will be meshed with `MESHPLN` times `MESHPLN` dummy quad elements.	I > 1

Remark

The default is sufficient in most cases.

MICRO

Micro-zoning Parameter

Defines the accuracy of the initial conditions in Eulerian elements, when using the geometrical shape definition

Format and Example	Default
`PARAM,MICRO,value`	10
`PARAM,MICRO,15`	

Option	Meaning	Type
`value`	Micro-zoning parameter	I > 0

Remarks

1. `MICRO3` is the number of micro zones into which an element is subdivided during initial condition generation.
2. The default MICRO = 10 results in material fractions as accurate as 0.001. If a higher accuracy is required, a greater value for `MICRO` can be used, but the CPU time for the generation increases rapidly.
3. Micro zoning is only used when the initial conditions of the Eulerian material are specified on a TICEUL entry.

MINSTEP

Minimum Time Step

Defines the minimum time step that causes the analysis to terminate.

Format and Example	Default
`PARAM,MINSTEP,value`	10% of INISTEP
`PARAM,MINSTEP,1.E-6`	

Option	Meaning	Type
`value`	When the time step is less than the MINSTEP value, the analysis terminates.	R > 0.0

Remarks

1. When the elements become very distorted, in a poorly designed mesh for example, or when they have endured a very large distortion, the time step may drop dramatically. The analysis continues, however, and a lot of computer resources may be wasted. This option allows you to specify a minimum time step that causes the analysis to terminate.
2. See Chapter 9: Running the Analysis, Terminating the Analysis for details on analysis termination.

MIXGAS

Controls Updating of Gas Fractions

Specifies whether the gas constants of the Euler material or of gas bags are updated based on the gas composition and temperature.

Format and Example	Default
`PARAM,MIXGAS,option`	NO
`PARAM,MIXGAS,YES`	

Option	Meaning	Type
`YES`	The gas constants for the Euler material and any gas bags are recalculated based on temperature and gas composition.	C
`NO`	Euler and gas bag gas constants are not recalculated.	C

Remarks

1. This parameter is only defined for use with GBAG gas bag definitions and/or the single-material Euler solver.
2. This parameter can be used in conjunction with INFLATR, INFLATR1, INFLHYB, and INFLHYB1 inflator definitions and with PORHOLE, PERMEAB, PORFGBG, and PERMGBG porosity definitions.
3. By default, `PARAM,MIXGAS` is set to `YES` if any INFLHYB, INFLHYB1, or INFLGAS entries are present.

NASIGN

Echo Ignored Data Entries

Toggles the echo of valid MSC Nastran and/or Dyna data entries that are ignored by Dytran.

Format and Example	Default
`PARAM,NASIGN,value`	`YES`
`PARAM,NASIGN,NO`	

Option	Meaning		Type
`value`	`YES`	Echo ignored entries.	C
	`NO`	Do not echo.	

Remarks

1. The echo of the ignored data entries is output to a file with the extension IGN.
2. Large input that originates from MSC Nastran or Dytran may produce a large amount of output and slow down the input processing.

NZEROVEL

Auto Constrain Failed Nodes

Sets the velocity of a node to zero in case all attached elements have failed.

Format and Example	Default
PARAM,NZEROVEL,option	NO
PARAM,NZEROVEL,YES	

Option	Meaning	Type
YES	Perform check and set the velocity to zero if all attached elements have failed.	C
NO	Do not perform check.	

Remarks

1. This parameter applies only to nodes of Lagrangian elements.
2. Specifying NO reduces the CPU overhead time.
3. When the velocity of a node is set to zero, effectively the node is constraint, like an SPC or SPC1.
4. Special attention is necessary for the contact definition. If the failed node is not taken out of the contact, it behaves as a rigid boundary constraint. Choose the appropriate METHOD for the SECACT entry on the CONTACT bulk data entry option.

OLDLAGTET Uses Collapsed Hexahedron Scheme for CTETRA

Activates the collapsed hexahedron scheme as default for Lagrangian CTETRA elements

Format and Example	Default
`PARAM,OLDLAGTET,value`	0
`PARAM,OLDLAGTET,1`	

Value	Meaning	Type
0	Default integration scheme for Lagrangian CTRETA elements will be the new linear tetrahedron FE one.	I
1	Default integration scheme for Lagrangian CTETRA elements will be the collapsed hexahedron scheme.	

Remark

The current default integration scheme for Lagrangian CTETRA elements use linear tetrahedron FE one. It is more consistent (in terms of accuracy) and efficient (both in memory and CPU time) compared with the collapsed hexahedron scheme. The old scheme based on collapsed hexahedron with reduced integration is deactivated. If the old scheme is activated, it is used as default. But, it is still possible to use the new scheme for CTETRA by using separate PSOLID with `IN = 1` and `ISOP = 1` combination.

OLDSHPL Uses old plastic calculation for shell elements

Activates old (pre Dytran 2019) plastic calculation instead of the new calculation that was introduced in Dytran 2019.

Format and Example	Default
`PARAM,OLDSHPL,option`	`NO`
`PARAM,OLDSHPL,YES`	

Option	Meaning	Type
`YES`	Use old calculation method.	C
`NO`	Use new calculation method.	

Remark

The plastic calculation for shell elements was updated in Dytran 2019. However, in order to get the same results as before, please use `PARAM,OLDSHPL,YES`.

Generally, the new option gives stiffer, but more accurate results than the old option.

PARALLEL

Parallel Execution Information

The option allows you to gather information on the parallel section of Dytran.

Format and Example
`PARAM,PARALLEL,option,value`
`PARAM,PARALLEL,INFPAR,[ON/OFF]`

Option	Meaning	Type	Default
`INFPAR`	A report is written on the actual amount of work done at the reported parallel levels.	C	OFF

Remarks

1. A summary on the parallel operation when using the shared-memory mode can be requested by including a `PARAM,PARALLEL,INFPAR,ON` entry in the input file. This request is not available in a restart of an analysis.
2. Currently, the information on the parallel sections is available for the shell solver only.

PLCOVCUT

Pressure Cut Off Time

Defines time when PLCOVER is cut off.

Format and Example	Default
PARAM,PLCOVCUT,value	0.0
PARAM,PLCOVCUT,3.E-3	

Option	Meaning	Type
value	If there are one or more COUPLE definitions with a PLCOVER specified on the COUOPT entry, a cut off is applied to the PLCOVER until time = PLCOVCUT.	R
	From time = 0 to time = PLCOVCUT, the PLCOVER is cut off to the pressure in the intersected Eulerian element.	
	For times greater than PLCOVCUT, the full PLCOVER is applied to the coupling (SUBSURF)SURFACE.	
	This parameter is useful in air-bag analyses, where PLCOVER is used to model the environment pressure. During the early stages of the deployment of the air bag, the pressure inside the bag may drop. Applying the full PLCOVER may lead to an unstable deployment of the air bag.	

Remark

See also the COUPLE and COUOPT Bulk Data entries.

PMINFAIL

Switches Failure at Spall Limit

Defines Lagrangian solid element failure on reaching the spall limit.

Format and Example	Default
`PARAM,PMINFAIL,option`	NO
`PARAM,PMINFAIL,YES`	

Option	Meaning	Type
YES	Element failure on spall limit	C
NO	No element failure on spall limit	

Remarks

1. Lagrangian elements (CHEXA) that have a material definition with a failure model will fail when the parameter is set to `YES` and the spall limit (minimum pressure) is reached, even when the other failure criterion is not yet reached.
2. The spall limit is set on the PMINC entry. (See also the DMATentry).

RBE2INFO

Lists MATRIG and RBE2 Grid Points

The grid points attached to MATRIG and RBE2assemblies are listed to the output file.

Format and Example	Default
`PARAM,RBE2INFO,option`	`GRIDOFF`
`PARAM,RBE2INFO,GRIDON`	

Option	Meaning	Type
`GRIDON`	Information is issued	C
`GRIDOFF`	No information is issued	

RHOCUT

Global Density Cutoff Value

Defines the minimum density for all Eulerian elements.

Format and Example	Default
`PARAM,RHOCUT,value`	See Remark 4.
`PARAM,RHOCUT,1.E-10`	

Option	Meaning	Type
`value`	Density cutoff	R > 0.0

Remarks

1. Any Eulerian element with a density less than `RHOCUT` is considered to be empty. All of its variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of element B after transport is less than `RHOCUT`, then no transport is done.
   b. If the density of element A after transport is less than `RHOCUT`, then all of the mass is transported to element B.
3. A reasonable value of `RHOCUT` is 1.E–5 times the initial density.
4. If only `RHOCUT` is defined, all Eulerian elements use the `RHOCUT` value as cutoff density. If `RHOCUT` is omitted, all Eulerian elements use a cutoff density automatically set to 1.E–5 times a characteristic density. For single-material Eulerian elements, this characteristic density is the reference density.

RJSTIFF

Rigid-joint Stiffness

Defines the stiffness of a rigid joint.

Format and Example	Default
`PARAM,RJSTIFF,value`	1.0
`PARAM,RJSTIFF,100.`	

Option	Meaning	Type
`value`	Multiplication factor for the stiffness of all rigid joints	R > 0.0

Remarks

1. The absolute stiffness of rigid joints is calculated automatically by Dytran. The stiffness of joints is taken so that a stable solution is guaranteed. The stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.
2. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken because too high a value may lead to an unstable calculation.

RKSCHEME

Runge-Kutta Time-Integration Scheme

Defines the type of time-integration scheme used in the Roe solver.

Format and Example	Default
`PARAM,RKSCHEME,number`	See Remark 1.
`PARAM,RKSCHEME,3`	

Option	Meaning	Type
`number`	Number of Runge-Kutta stages.	I > 0

Remarks

1. This parameter can only be used in combination with `PARAM,LIMITER,ROE`. The default number of stages depends on the spatial accuracy of the solution scheme. One stage is used for first order spatial accuracy, and three stages for second order spatial accuracy.
2. For more details on the Runge-Kutta time-integration scheme see *Dytran Theory Manual*, Chapter 6: Standard Euler Solver.

ROHYDRO

Density Cut-Off Value

Defines the minimum density for hydrodynamic, single-material Eulerian elements.

Format and Example	Default
`PARAM,ROHYDRO,value`	See Remark 3.
`PARAM,ROHYDRO,1.E-6`	

Option	Meaning	Type
`value`	Density cutoff.	R > 0.0

Remarks

1. Hydrodynamic, single-material Eulerian elements with a density less than `ROHYDRO` are considered to be empty. All of the variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of element B after transport is less than `ROHYDRO`, then no transport is done.
   b. If the density of element A after transport is less than `ROHYDRO`, then all of the mass is transported to element B.
3. By default, the cutoff density for hydrodynamic Eulerian elements is set to 1.E–5 times the material reference density.

ROMULTI

Density Cut-Off Value

Defines the minimum density for multimaterial Eulerian elements.

Format and Example	Default
`PARAM,ROMULT,value`	See Remark 3.
`PARAM,ROMULT,1.E-6`	

Option	Meaning	Type
`value`	Density cutoff	R > 0.0

Remarks

1. Multimaterial Eulerian elements with a density less than `ROMULTI` are considered to be empty. All of the variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of a specific material in element B after transport is less than `ROMULTI`, no transport is done.
   b. If the density of a specific material in element A after transport is less than `ROMULTI`, all of the mass of that material is transported to element B.
3. By default, the cut-off density is set for each material separately as 1.E–5 times the material reference density.

ROSTR

Density Cut-Off Value

Defines the minimum density for single-material Eulerian elements with shear strength.

Format and Example	Default
`PARAM,ROSTR,value`	See Remark 3.
`PARAM,ROSTR,1.E-6`	

Option	Meaning	Type
`value`	Density cutoff	R > 0.0

Remarks

1. Single-material Eulerian elements with shear strength with a density less than `ROSTR` are considered to be empty. All of the variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of element B after transport is less than `ROSTR`, then no transport is done.
   b. If the density of element A after transport is less than `ROSTR`, then all of the mass is transported to element B.
3. By default the cut-off density for Eulerian elements with shear strength is set to 1.E–5 times the material reference density.

RSTDROP

Type of Elements Dropped at Restart

Defines the type of elements to be removed from the calculation when restarting an analysis.

Format and Example	Default
`PARAM,RSTDROP,option`	No elements are dropped.
`PARAM,RSTDROP,LAGRANGE`	

Option	Meaning	Type
`LAGRANGE`	Lagrangian solid elements are dropped.	C
`EULER`	Eulerian elements are dropped.	
`MEMBRANE`	Membrane elements are dropped.	
`SURFACE`	Rigid bodies and coupling surfaces are dropped.	

Remarks

1. All elements of the specified type are removed from the calculation. It is not possible to drop a part of a Eulerian or Lagrangian mesh.
2. If Lagrangian solids or membranes are dropped from a coupled calculation, the surfaces should also be dropped to prevent surfaces being present that are not attached to anything.
3. The `EULER` option only works for a Eulerian mesh containing a single hydrodynamic material.

SCALEMAS

Mass Scaling Definition

Parameters

Defines the activation of mass scaling.

Format and Example	Default
PARAM,SCALEMAS,DTMIN,MXPERC,STEPS	See Remark 3.
PARAM,SCALEMAS,1E-6, 100.0,1	

Option	Meaning	Type
DTMIN	Minimum allowable time step	R See Remarks 1. and 7.
MXPERC	Maximum percentage of added numerical mass with respect to original mass.	R ≥ 0.0 See Remarks 1. and 6.
STEPS	Number of steps	I > 0 See Remarks 2. and 6.

Remarks

1. Numerical mass will be added to any Lagrangian element (solid, triangular, quadrilateral, rod, bar, and beam), such that the time step of the element never becomes less than:

   dt = STEPFCT*DTMIN.

   where

dt	=	timestep of calculation
STEPFCT	=	timestep safety factor (see PARAM, STEPFCT)
DTMIN	=	value specified on the PARAM, SCALEMAS entry

   If the added mass of a certain element exceeds the maximum percentage (MXPERC) of its original mass, no more mass will be added, and subsequently, the time step may decrease again.

2. The value of STEPS determines the checking frequency against the mass scaling criterion; the check is done for every defined number of STEPS. STEPS = 1 is recommended.
3. The values for DTMIN, MXPERC, and STEPS are required to be defined.
4. By requesting MSMASS in an output request, the ratio of scaled mass to the original mass of the elements can be retrieved. By making fringe plots of this parameter, a check can be made if the mass has not been added in a critical area.

5. See *Dytran User's Guide*, Chapter 8: Prestress Analysis and Example Input Data, Mass Scaling Definition for instructions on how to use this entry.
6. DTMIN can be negative only when PARAM, SMSOPT is used. In this case, both of conventional mass scaling and selective mass scaling are used. Absolute value of negative DTMIN is the time step increment of conventional mass scaling. MXPERC and STEPS will be ignored.
7. DTMIN can be a negative value when SMS (OPTION=1, 2, 3, or 4) is used. A positive number is the target time step and a negative number means that the absolute value is the factor of time step increment. For example:

   ```
   PARAM,SCALEMAS,-40.0,,,3,10001,1
   ```

   Target time step = 40.0 x (time step of model)

SHELLFORM

Sets the Default of the Shell Formulation

Sets the default for the shell formulation for quadrilateral elements.

Format and Example	Default
PARAM,SHELLFORM,option	KEYHOFF
PARAM,SHELLFORM,BLT	

Option	Meaning	Type
BLT	The shell-formulation default is BLT.	C
KEYHOFF	The shell-formulation default if KEYHOFF.	

Remarks

1. The PARAM,SHELLFORM changes the default formulation for quadrilateral shell elements. All shell properties entries that do not explicitly define the formulation, use the default as specified on the PARAM entry.
2. Triangular shell elements have only one formulation (C0-TRIA). Therefore, the PARAM is ignored for triangular elements.
3. For more information, see also *Dytran User's Guide*, Chapter 5: Application Sensitive Default Setting.

SHELLKEY

Sets the KEYHOFF Shell Formulation

Sets the default of the KEYHOFF shell formulation.

Format and Example	Default
`PARAM,SHELLKEY,option`	`New`
`PARAM,SHELLKEY,OLD`	

Option	Meaning	Type
`NEW`	Use new KEYHOFF element formulation.	C
`OLD`	Use old KEYHOFF element formulation.	

Remarks

1. In Dytran 2019, the enhancement was updated related to KEYHOFF shell formulation. Because of the enhancement, the results will be different if the large rotation is applied on the element with KEYHOFF formulation during simulation. If a user wants to get the old results, please set this PARAM with OLD option.

SHELMSYS

Shell Element System Definition

Defines the shell element system for the BLT shells.

Format and Example
`PARAM,SHELMSYS,option`
`PARAM,SHELMSYS,SIDE21`

Option	Meaning	Type	Default
`SIDE21`	x-axis along side21	C	MIDSIDES
`MIDSIDES`	x-axis connecting midpoints		

Remarks

1. `SIDE21` puts the x-axis along side21 of the element, whereas `MIDSIDES` puts the x-axis along the vector connecting the midpoints of the side14 and side32.
2. Using the `SIDE21` option for the BLT shell will result in the same Belytschko-Lin-Tsay implementation as BELY.

SHPLAST

Type of Plane-Stress Plasticity for Shells

Specifies the type of calculation used to determine the plane-stress plasticity method for shells.

Format and Example	Default
PARAM,SHPLAST,option	ITER
PARAM,SHPLAST,VECT	

Option	Meaning	Type
RADIAL	Noniterative, approximate radial return.	C
VECT	Iterative, vectorized with three iterations.	
ITER	Nonvectorized iterations.	

Remarks

1. The RADIAL approach does not require iterations and, therefore, is the most efficient. It is, however, an approximation.
2. The other two approaches iterate to find the solution. ITER is the best since it takes as many iterations as are necessary. On vector machines, such as CRAY, this is inefficient since it cannot be vectorized. VECT always performs three vectorized iterations, which is more efficient. However, three iterations may not be enough, and inaccuracies could occur.
3. For more information, see *Dytran User's Guide*, Chapter 5: Application Sensitive Default Setting.

SHSTRDEF Composite Shell Stress and Strain Output Definition

Specifies the default coordinate system for the stress and strain output of composite shells.

Format and Example
`PARAM,SHSTRDEF,option`
`PARAM,SHSTRDEF,ELEM`

Option	Meaning	Type	Default
`FIBER`	Stresses and strains are output in the fiber and matrix directions.	C	FIBER
`ELEM`	Stresses and strains are output in the element coordinate system.		

Remark

The default setting can be overruled per property on a PCOMPA entry on the `STRDEF` field.

SHTHICK

Shell-Thickness Modification Option

Specifies whether or not the thickness of the shell changes with membrane straining.

Format and Example	Default
PARAM,SHTHICK,option	YES
PARAM,SHTHICK,YES	

Option	Meaning	Type
YES	Shell thickness is modified according to the membrane strain.	C
NO	Shell thickness is constant.	

Remarks

1. The YES option gives a true large-strain shell but requires some extra computation.
2. The NO option should give adequate results as long as the membrane strains are not very large (i.e., not more than 5–10%).
3. This option applies to all the formulations of the shell elements, except for the PCOMP. The thickness of PCOMP shell elements will always remain constant.
4. For more information, see *Dytran User's Guide*, Chapter 5: Application Sensitive Default Setting.

SLELM

Store Shell Sublayer Variables

Defines whether shell sublayer variables are to be stored in the element arrays.

Format and Example	Default
PARAM,SLELM,option	YES
PARAM,SLELM,NO	

Option	Meaning	Type
YES	Store as an element variable.	C
NO	Do not store as an element variable.	

Remarks

1. This parameter applies only to shell elements.
2. The shell sublayer variables are primarily stored in sublayer arrays. They can be copied into the element arrays only for specific output purposes.
3. Specifying NO reduces the CPU overhead time.
4. Irrespective of the entry on this parameter, sublayer variables are accessible in the sublayer arrays.

   For example, requesting TXX1 retrieves the stress from the element array, whereas TXX01 retrieves it from the sublayer arrays.
5. MSC Nastran initialization always causes SLELM = YES.

SMF

Setting for sheet metal forming simulation

Defines default setting for sheet metal forming simulation.

Format and Example	Default
PARAM,SMF,OPTION1,OPTION2,DIR,NSUBCY,ITABLE	OPTION1=OFF OPTION2=OFF
PARAM,SMF,ON,ON,Z,2	

Option	Meaning	Type
OPTION1	Usage of contact algorithm for sheet metal forming ON contact algorithm for sheet metal forming is used. OFF contact algorithm for sheet metal forming is not used.	C, See Remark 1.
OPTION2	Usage of time step algorithm to get the stable condition under severe deformation ON the time step algorithm is used. OFF the time step algorithm is not used.	C, See Remark 5.
DIR	Moving direction of tools of sheet metal forming simulation. Only one of X, Y, and Z is allowed. Only available when OPTION1 = ON. default=not used	C
NSUBCY	Number of cycles to skip contact calculation. default=not used.	I ≥ 0, See Remark 2.
ITABLE	Table id to control NSUBCY. x value in the table is time x and y value in the table is step size for skipping contact calculation. default=not used	I ≥ 0, See Remark 2.

Remarks

1. When OPTION1=ON is used, only sheet metal is a flexible body and all other tools are rigid bodies. Tools must be set in primary contact bodies in the contact definition. In addition, all tools have only a translational degree of freedom. Only for this case, the faster contact algorithm will give correct answers.
2. Only one of NSUBCY and ITABLE is available. NSUBCY is used to skip contact calculation until the end of the simulation. When ITABLE is used, y-value is rounded off. When the y value which is rounded off is less than 1 means subcycling is off. When it is larger than 1, it is set to the same value as NSUBCY at the time which is set in x value.
3. PARAM,CONTACT,STIFF,SLSTIFF is only available with OPTION1=ON in PARAM,SMF.

4. When `OPTION1=ON` is set, the default value of `PARAM,LIMCUB` is set to 100,000. If you want to change the value, please check `PARAM,LIMCUB`.
5. For getting stable results under severe deformation such as wrinkling, different characteristic length calculation is used for quadrilateral elements.

   `OPTION2 = OFF`:

$$\text{Characteristic length} = \frac{\text{Area}}{\max(\text{length of side1}, \text{length of side2}, \text{length of side3}, \text{length of side4})}$$

   `OPTION2 = ON`:

$$\text{Characteristic length} = \frac{\text{Area}}{\max(\text{diagonal length from grid 1 to 3}, \text{diagonal length from grid 2 to 4})}$$

SMP, ,BATCHSIZE SMP Batchsize and CPULOOP Definition

Define batch size and number of CPU loops, per entity type.

Format and Example
PARAM,SMP,<entity>,BATCHSIZE,<value1>,CPULOOPS,<value2>
PARAM,SMP,BEAMS,BATCHSIZE,AUTOMATIC
PARAM,SMP,BEAMS,BATCHSIZE,256

Option	Meaning		Type
ENTITY	Define the batchsize for the following entities:		C; required
	BEAMS	beam elements	
	SHELLS	shell elements	
	SOLIDS	solid elements	
	CONTACT	contacts	
VALUE1	See Remarks 1. and 3..		C, I>0 (default = 63)
	The batchsize for this entity (Integer number)		
	The keyword AUTOMATIC (character)		
VALUE2	If value1 is defined as AUTOMATIC, the user may define the number of loops per CPU (DEFAULT = 2). See Remark 2..		I>0, not required

Remarks

1. The batchsize may be defined per entity type. The SMP parts of Dytran uses the batchsize to assign a certain number of batches to each CPU. Example:
   - Batchsize = 63 (default)
   - Number of Beam elements = 580
   - Number of CPUs = 4

   As a results, Dytran:
   - Needs to process 10 batches (9x63 + 1x13)
   - To spread this evenly over four CPUs, Dytran increases this to 12 batches because then each CPU can process 3 batches.
   - As a result, the batch size is recalculated for 12 batches to: batchsize = 49
   - This means that:

     CPU# 1,2,3 process three batches of 49 elements each (3x3x49 = 441 elements)

     CPU# 4 processes two batches of 49 elements and one batch of 41

It is clear that the user-defined batchsize may not be used ultimately. The reason for this is that each CPU is allocated the same number of batches in order to optimize the work done by each processor.

2. When the batchsize is set to AUTOMATIC, Dytran computes an initial batchsize based on the number of batches processed by each CPU. By default, Dytran processes two batches per CPU. In the example above, this means that based on:

   - Batchsize = AUTOMATIC
   - Number of batches per CPU = 2 (default)
   - Number of Beam elements = 580
   - Number of CPUs = 4

   As a result, Dytran:

   - Computes an initial batchsize of 72 (=580/(4*2))
   - Increases this batchsize to a higher number to make sure the number of batches fits in the fixed number of CPU's: batchsize = 72 + (580 – 4*2*72) = 72 + 4 = 76 elements
   - Based on this initial batchsize, the number of batches would be: 1 + (580-1)/76 = 8 batches
   - This fits nicely on four CPU's as expected
   - The CPUs process:

     CPU# 1,2,3 processes two batches of 76 elements each (3x2x76 = 456 elements)
     CPU# 4 processes one batch of 76 elements and 1 batch of 48 elements.

3. By default, Dytran does not use the AUTOMATIC batchsize algorithm. Testing has shown that an initial batchsize of 63 yields the fastest results. This may vary depending on the problem simulated and the number of elements and may be worth adjusting.

   The reason for this may be found in the fact, that with many elements in the model, the number of elements per batch increases significantly when the AUTOMATIC batchsize algorithm is used. Each CPU will have to process two batches (by default) with many elements. Based on the cache memory, this is usually not optimal for a CPU processor. Shorter batchsizes (< 128) will do a better job.

SMP, ,CPUINFO

SMP CPU Information

Define the CPU information per entity type

Format and Example
PARAM,SMP,<entity>,CPUINFO,<value1>
PARAM,SMP,CONTACT,CPUINFO,ON
PARAM,SMP,FSCOUP,CPUINFO,ON

Option	Meaning		Type
ENTITY	Define the CPUINFO for the following entities:		C; required
	BEAMS	beam elements	
	SHELLS	shell elements	
	SOLIDS	solid elements	
	CONTACT	contacts	
	FORGE2	N/A	
	SF	N/A	
	MEMBR	membrane elements	
	EULT9	Eulerian Roe Solver	
	FSCOUP	Fast coupling algorithm	
	ALL	Output for all of the above	
VALUE1	ON	Ask for parallel CPU time information	C
	OFF	No CPU time information (default)	

Remark

By default, Dytran does not give any SMP CPU timings per entity. In order to judge the efficiency of the different algorithms in Dytran, this parameter may be used to judge the speedup acquired by processing on multiple CPUs.

SMSOPT

Time step scaling factor for selective mass scaling

Defines portion of time step scaling factor for selective mass scaling when both the conventional mass scaling and selective mass scaling are used.

Format and Example	Default
`PARAM,SMSOPT,FACTOR,OPTION,SID,THREAD`	`Not used.`
`PARAM,SMSOPT,-3.0,3,1001,2`	

Option	Meaning	Type
`FACTOR`	Factor of time step for selective mass scaling. `> 0.0` Target time step `< 0.0` Target time step increment factor	R. See Remark 1. and 2.
`OPTION`	Usage of mass scaling 1: translational mass scaling without diagonal scaling in sms matrix 2: translational/rotational mass scaling without diagonal scaling in sms matrix 3: translational mass scaling with automatic diagonal scaling in sms matrix 4: translational/rotational mass scaling with automatic diagonal scaling in sms matrix	I ≥ 0, required. See Remark 3.
`SID`	`SET1` Id which includes property id's to be used for selective mass scaling	I ≥ 0, required. See Remark 3.
`THREAD`	Activating SMP capability for Pardiso solver. default=ON `ON` Activate SMP capability `OFF` Deactivate SMP capability	C. See Remark 5.

Remarks

1. `FACTOR` can be a negative value. A positive number is the target time step and a negative number means that the absolute value is the factor of time step increment.

   For example: `PARAM,SMSOPT,-40.0,3,10001,ON` Target time step = 40.0 x (time step of model)

2. When a negative value is used in `FACTOR`, negative `DTMIN` in `PARAM,SCALEMAS` can be used. When both are used the combination of selective mass scaling and conventional mass scaling is used.

3. Selective mass scaling does not scale the rigid body mass and is, therefore, more accurate. Since it is memory and CPU intensive, it should be applied only to deformable elements.
4. Selective Mass Scaling (SMS) can reduce the number of steps and high-frequency dynamic responses. Since the linear equation is required to be solved at each time step, it takes longer than the calculation of each step.

   Currently, Selective Mass Scaling capability has the following limitations:

   - SMS is only available for shell elements (quad and triangular) in a flexible body.
   - SMS is only applicable for a free body or a body with fixed nodal boundary conditions. All other enforced motions are not available in SMS.
   - SMS is available in flexible bodies with forces/initial velocities.
   - SMS is not available for certain parts of flexible bodies. For SMS, all flexible bodies must be assigned in `SET1` definition and the flexible body must be separated from rigid bodies.
5. Calculation of selective mass scaling is done in each degree of freedom independently. SMP capability can be used for selective mass scaling.

   When `THREAD` is set to `OFF`, 1 thread will be used for all `OPTION`s.

   When `THREAD` is set to `ON`, 3 threads will be used for `OPTION = 1` and `3`, and `6` threads will be used for `OPTION = 2` and `4`.

SNDLIM

Sound Speed Minimum Value

Defines the minimum value for the speed of sound.

Format and Example	Default
PARAM,SNDLIM,value	1.E-3
PARAM,SNDLIM,1.E-6	

Option	Meaning	Type
value	Minimum value of speed of sound.	R > 0.0

Remarks

1. This parameter is used to avoid the possibility of division by zero in the time step calculation.
2. SNDLIM has the units of velocity.

SPREMAP 1-D Spherical Symmetric Euler Archive Remap

Allows a 1-D spherical symmetric Euler archive importation to a 3-D simulation.

Format and Example
`PARAM,SPREMAP,X0,Y0,Z0,range`
`PARAM,SPREMAP 0.0 0.5 0.5`

Option	Meaning	Type	Default
`X0, Y0, Z0`	X, Y, Z coordinate of the point at which the 1-D spherical symmetric mesh is remapped.	R	0.0
`RANGE`	Only material whose distance to (X0,Y0,Z0) is smaller than "range" will be initialized with the 1-D spherical symmetric Euler archive.	R	1e+20

Remarks

1. Since 1-D spherical symmetric simulations run much faster than 3-D simulation, it can save much CPU time to do the first part of the simulation with a 1-D axial symmetric mesh. Afterwards, the 1-D spherical symmetric Euler archive is imported into the 3-D simulation. By default, the 1-D spherical symmetric archive will not be expanded in 3-D. To enable this expansion, `PARAM,SPREMAP` has to be used. It is useful for blast wave simulations. The 1-D spherical symmetric simulation has to be terminated before the blast wave approaches any 3-D structure.
2. This import of Euler archives is done by means of the `EID` option in the pth file.
3. To generate axial symmetric meshes `PARAM`,SPHERSYM can be used.
4. The remap of 1-D spherical symmetric to axial symmetric Euler meshes is supported.

SPHERSYM

Spherical Symmetric Analyses

Enables an efficient and accurate 1D spherical symmetric solution for Eulerian materials. A much larger time step becomes possible by basing the time step only on the mesh-size in radial direction.

Format and Example	Default
`PARAM,SPHERSYM,MESHTYPE,RADAXIS,PHI`	
`PARAM,SPHERSYM,RECT,X,2`	`PHI: 0`

Option	Meaning		Type
`MESHTYPE`	Two types of Euler meshes are supported:		`C: [SPHERIC,RECT]`
	Axial symmetric meshes:	`MESHTYPE = SPHERIC`	
	Rectangular meshes:	`MESHTYPE = RECT`	
`Radial Axis`	X Y Z		`C: [X,Y,X]`
`PHI`	Only used for MESHTYPE = RECT.		R > 0.0
	Used to creates a 1d Spherical mesh with angles +PHI/2 and –PHI/2		

Remarks

1. Only available for Eulerian elements and does not support Lagrange elements. The effect of this parameter is not limited to the solvers. Also Euler archives will reflect the modified Euler mesh geometry.
2. The Euler mesh can already be symmetric but also a rectangular mesh comprising of one row of elements can be used. Using the angle specified by PHI this Euler mesh is mapped into a 1D spherical symmetric mesh.
3. The Euler mesh has to consist of one row of elements.
4. In the time step computation only the mesh-size in radial direction will be taken into account.

STEPFCT

Time Step Scale Factor

Defines a scale factor to be used on the internally calculated time step.

Format and Example	Default
`PARAM,STEPFCT,value`	0.666
`PARAM,STEPFCT,0.5`	

Option	Meaning	Type
`value`	Time-step safety factor	$0.0 < R \leq 1.0$

Remarks

1. The actual time step used in Dytran is the product of the internal time step and the time step safety factor.
2. The default value works well in the majority of situations and gives an efficient solution while maintaining a stable solution.
3. In a calculation with a coupling surface,FBLEND must be greater than or equal to `STEPFCT` to avoid instabilities (see `PARAM,FBLEND`).
4. For many calculations, `STEPFCT` can be set to 0.9, unless you are running a problem that has coupling surfaces defined.
5. A different parameter can be used to set the time step safety factor for Lagrangian elements (see `PARAM, STEPFCTL`).

STEPFCTL Time-step Scale Factor for Lagrangain Elements

Defines a scale factor to be used on the internally calculated time step of Lagrangian elements only.

Format and Example	Default
`PARAM,STEPFCTL,value`	`STEPFCT`
`PARAM,STEPFCTL,0.9`	

Option	Meaning	Type
`value`	Time-step safety factor	$0.0 < R \leq 1.0$

Remarks

1. For many calculations, `STEPFCTL` can be set to 0.9
2. `PARAM,STEPFCT` can be used to set the time step safety factor for all elements (see `PARAM,STEPFCT`). The default for `STEPFCT` is 0.666
3. The actual time step used in Dytran is the product of the internal time step and the time-step safety factor. If `STEPFCTL` is used, the Lagrangian elements will use this value for the safety factor. The Eulerian elements will still use `STEPFCT`. The timestep used in Dytran is the minimum value of the two products. To this end, an efficient solution, while also maintaining a stable solution, is achieved.

STRNOUT

Shell Sublayer Strain Output

Saves the total strains and equivalent effective stress (von Mises stress) at shell sublayers for output.

Format and Example	Default
PARAM,STRNOUT,option	YES
PARAM,STRNOUT,YES	

Option	Meaning	Type
YES	Save	C
NO	Do not save.	

Remarks

1. A limited output set saves memory.
2. Perfectly elastic materials only have the limited output set.
3. Total strain output for shell composite materials can be requested from the PCOMPA Bulk Data entry.

TOLCHK

Tolerance for Fast Coupling

Fast coupling requires that Euler elements are orthogonal. This means that normals of Euler element faces have to be in a coordinate direction. In practice there can be small errors in the geometry of the Euler element and Euler face normals do not always exactly point into a coordinate direction. Small deviations from the coordinate direction do not give problems and are allowed by fast coupling.

To check the direction of an Euler face, the face normal vector is projected onto the closest coordinate direction. If this projection is 1, the normal is exactly in the coordinate direction. When this projection is within a sufficient small tolerance of 1, the face can be handled by fast coupling. The tolerance used is **TOLCHK**. If the projection is smaller than **1-TOLCHK**, the face cannot be handled by fast coupling and the analysis terminates. Options are then to slightly increase **TOLCHK** , write out double precision format in PATRAN, use general coupling, or use the **MESH** entry. Increasing **TOLCHK** too much can make the coupling surface computation less accurate. To keep the computation accurate, the maximal allowed value of **TOLCHK** is 1e-6. If `PARAM,TOLCHK` is not used, the tolerance used is 1e-14.

Format and Example	Default
`PARAM,TOLCHK,value`	None
`PARAM,TOLCHK,1e-10`	

Option	Meaning		Type
`value`	0	Tolerance in accepting faces by fast coupling.	R > 0.0

TOLFAC Increase the Projection Tolerance for CONTACT at Initialization

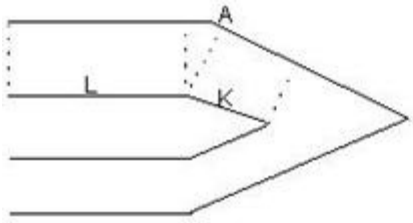
This parameter is a scale factor for the values of TOLPR1 of the CONTACT options that use BPLANE option.

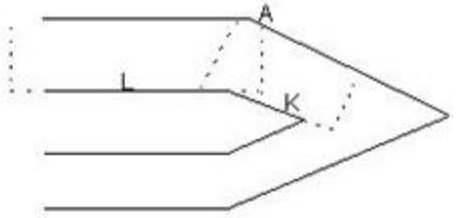
Format and Example	Default
PARAM,CONTACT,TOLFAC,value	1.
PARAM,CONTACT,TOLFAC,1000.	

Option	Meaning	Type
value	Scale factor for TOLPR1	R > 0.0

Remarks

1. This parameter is important for initialization of BPLANE contact. The faces of the contact surface are enlarged with a value of TOLPR1. However, this might not be enough when the air bag is offset folded. On the other hand, a large value of TOLPR1 might induce hooking. Therefore a new parameter is introduced called TOLFAC. The value of TOLFAC scales the value of TOLPR1 only at initialization, such that the contact is correctly found.
2. A correct value of TOLFAC can be up to 1000. or even higher.





UGASC

Universal Gas Constant

Defines a value for the universal gas constant.

Format and Example	Default
`PARAM,UGASC,value`	Required
`PARAM,UGASC,8.3144`	

Option	Meaning	Type
`value`	Value of the universal gas constant	R > 0.0

Remarks

1. This entry must be used if the molar weight is used on an INITGAS entry, or if molar gas fractions are given on an IINFLFRAC entry.
2. Specify only one universal gas constant per problem.
3. In SI units, R equals 8.3145 J mol^{-1} K^{-1}.

   Using the tonne, mm, s system of units R has a value of 8314.5 tonne mm^2 s^{-2} mol^{-1} K^{-1}.

   In imperial units, R equals 1.9859 Btu $lbmol^{-1}$ $^oR^{-1}$ or 1545.3 ft lbf $lbmol^{-1}$ $^oR^{-1}$.

VARACTIV

Activation or Deactivation of Grid-Point, Element, or Face Variables

Grid point, element, or face variables are activated or deactivated by the Bulk Data entry, as well as the definition of new user variables. The name of the variable is redefined as well.

Format and Example
`PARAM,VARACTIV,(elname),(GEF),(varname),(datatype),(ACTDEAC),(newname)`
`PARAM,VARACTIV,LAGSOLID,ELEM,MASS,FLT,ACTIVE,ELMASS renames the variable MASS to ELMASS.`
`PARAM,VARACTIV,LAGSOLID,ELEM,NEWVAR,FLT,ACTIVE,NEWVAR creates a new variable NEWVAR as an element float value.`

Option	Meaning		Type	Default
`elname`	Name of the element:		C	Required
	`ELEM1D`	One-dimensional elements		
	`SHTRIA`	Triangular shell		
	`SHQUAD`	Quadrilateral shell		
	`MEMTRIA`	Membrane		
	`DUMTRIA`	Triangular dummy element		
	`DUMQUAD`	Quadrilateral dummy element		
	`LAGSOLID`	Lagrangian solid		
	`EULHYDRO`	Eulerian hydrodynamic solid		
	`EULSTRENGTH`	Eulerian solid with stress tensor		
	`MULTIEULHYDRO`	Multimaterial Eulerian solid		
	`ALL`	Activate all variables		
	`ALLPRINT`	Activate all variables and print a summary		
`GEF`	Entity name:		C	ELEM
	`GRID`	Grid point		
	`ELEM`	Element		
	`FACE`	Face		
`varname`	Name of the variable		C	Required
`datatype`	Data type of the variable:		C	FLT
	`INT`	Integer		
	`FLT`	Float		

Option	Meaning		Type	Default
	CHAR	Character		
ACTDEAC	Activate or deactivate variable:		C	ACTIVE
	ACTIVE	Activate a variable.		
	DEACTIVE	Deactivate a variable. (See Remark 4.)		
newname	Redefined name of the variable. (See Remark 3.)		C	Required

Remarks

1. The ALL entry activates all variables for all elements, regardless of whether they are used or not. The ALLPRINT prints a summary of all element variables, regardless whether there are any elements of a certain type or not. If ALL or ALLPRINT is entered, then no subsequent entries are required.
2. All entries are required if ALL or ALLPRINT are not specified, except for newname, which defaults to the original name.
3. When a variable is renamed, all subsequent references must be made to the new name; e.g., in output requests.
4. The deactivate option is potentially dangerous, since some options may require the variable in an indirect way. It is advised, therefore, not to deactivate standard Dytran variables.
5. This PARAM entry is a convenient way to introduce new variables to an entity, which can be used in user subroutines. The new variables is written to restart files and can be requested for output.
6. In the print file of an Dytran run, a summary is given for all variables of the element types that are used in the calculation (except when ALLPRINT was specified). The format of the printout is:

   ###-(CHAR)-NAME

   where

###	=	the variable ident number when the variable is active. If the variable has been deactivated, ### is printed as <->
CHAR	=	N if the variable is standard Dytran R if the variable has been renamed C if the variable has been defined by the user E if the variable is used for editing only
NAME	=	the (new) name of the variable

VDAMP

Dynamic Relaxation Parameter

Controls the global damping in the dynamic relaxation.

Format and Example	Default
`PARAM,VDAMP,value/option`	0.0
`PARAM,VDAMP,0.001`	
`PARAM,VDAMP,OFF`	

Option	Meaning	Type
`value`	Dynamic relaxation parameter	R ≥ 0.0
`OFF`	No dynamic relaxation	C

Remarks

1. The dynamic relaxation parameter is connected to the system natural frequency, ω, as $\beta = \varepsilon\omega\Delta t$, where ε denotes a percentage of critical damping. The damping occurs by factoring the velocities every time step as follows:

   $$F_1 = (1-\beta)/(1+\beta)$$

   $$F_2 = 1/(1+\beta)$$

   $$v^{n+1/2} = F_1 v^{n-1/2} + F_2 a^n \Delta t^n$$

   where v is the velocity, a is the acceleration, and β is the dynamic relaxation parameter.

2. At the restart of an analysis with dynamic relaxation, the dynamic relaxation can be switched off by `PARAM,VDAMP` or `PARAM,VDAMP,OFF`.

3. For a more comprehensive description of dynamic relaxation, see *Dytran Theory Manual*, Chapter 4: Models, Dynamic Relaxation.

VELCUT

Velocity Cutoff

Defines the minimum velocity.

Format and Example	Default
`PARAM,VELCUT,value`	1.E–6
`PARAM,VELCUT,1.0E-6`	

Option	Meaning	Type
`value`	Minimum velocity	R > 0.0

Remark

Any velocity less than `VELCUT` is set to zero. It is mainly used to eliminate harmless but annoying small values of velocity caused by round-off error and numerical dispersion.

VELMAX

Maximum Velocity

Defines the maximum velocity in Eulerian and Lagrangian meshes.

Format and Example	Default
PARAM,VELMAX,value,YES/NO	1.E10,YES
PARAM,VELMAX,1.E6	

Option	Meaning	Type
value	Maximum velocity	R > 0.0
YES	Remove the mass in Eulerian elements in which the velocity exceeds the maximum specified velocity.	C
NO	Do not remove the mass in Eulerian elements in which the velocity exceeds the maximum specified velocity.	

Remarks

1. For Eulerian elements, VELMAX is applied to the velocity components separately. Therefore, the magnitude of the velocity vector is limited to $\sqrt{3}$ VELMAX. For Lagrangian elements, all components are scaled down by the same factor to ensure that the maximal magnitude of the velocity vector does not exceed VELMAX.
2. Although it is not usually necessary to limit the velocity in Eulerian meshes, there are occasions in regions of near-vacuous flow for example, where specifying a maximum velocity can be advantageous. The same applies to Lagrangian meshes in contact regions for example. This parameter should be used with care.
3. Because very high velocities occur mostly in Eulerian elements with very small mass, the mass in these elements can be removed to keep the analysis stable. This option is not available for Lagrangian solid elements.
4. VELMAX must be greater than the minimum velocity specified by PARAM,VELCUT.
5. VELMAX applies the same maximum velocity to both translational and angular velocities.
6. VELMAX is not applied to rigid bodies defined by the RIGID entry, the MATRIG, and RBE2-FULLRIG entries.

VELMAX1 Maximum Translational and Angular Velocity

Defines the maximum translational and angular velocity in Eulerian and Lagrangian meshes.

Format and Example	Default
PARAM,VELMAX1,VELMAXT,YES/NO,VELMAXA	1.E20,YES,1.E20
PARAM,VELMAX1,1.E6	

Option	Meaning	Type
VELMAXT	Maximum translational velocity	R > 0.0
YES	Remove the mass in Eulerian elements in which the velocity exceeds the maximum specified velocity.	C
NO	Do not remove the mass in Eulerian elements in which the velocity exceeds the maximum specified velocity.	
VELMAXA	Maximum angular velocity	R > 0.0

Remarks

1. For Eulerian elements, VELMAXT is applied to the velocity components separately. Therefore, the magnitude of the velocity vector is limited to VELMAXT. For Lagrangian elements, all components are scaled down by the same factor to ensure that the maximal magnitude of the velocity vector does not exceed VELMAXT.
2. VELMAXA is used to limit the angular velocity of grid points. By default, the angular velocities are not limited. In general, there is no need to limit angular grid point velocities.
3. Although it is not usually necessary to limit the velocity in Eulerian meshes, there are occasions in regions of near-vacuous flow for example, where specifying a maximum velocity can be advantageous. The same applies to Lagrangian meshes in contact regions for example. This parameter should be used with care.
4. Because very high velocities occur mostly in Eulerian elements with very small mass, the mass in these elements can be removed to keep the analysis stable. This option is not available for Lagrangian solid elements.
5. VELMAXT must be greater than the minimum velocity specified by PARAM,VELCUT.
6. VELMAX1 is not applied to rigid bodies defined by the RIGID entry, the MATRIG, and RBE2-FULLRIG entries.
7. PARAM, VELMAX, and PARAM,VELMAX1 can be used in the same input deck. The value of PARAM,VELMAX will overrule the missing values of PARAM,VELMAX1.

   For example when using

   ```
   PARAM,VELMAX,1000,YES
   PARAM,VELMAX1,,YES,1e+10
   ```

the OUT file shows:

```
%I-INIT  02-p3_set_velmaxat, , ,(13),
SUMMARY VELMAX SETTINGS:
MAXIMAL VALUE TRANSLATIONAL VELOCITY = 0.100000E+04
MAXIMAL VALUE ANGULAR VELOCITY = 0.100000E+11
REMOVE EULERIAN MASS WHEN VELOCITY EXCEEDS LIMIT: YES
```

VISCOPLAS

Use Overstress Formula to Update Strain-rate Dependent Plasticity

Activate the overstress formula to update strain-rate dependent plasticity. This formula is normally used for viscous-plastic material.

Format and Example	Default
`PARAM,VISCOPLAS,value`	0
`PARAM,VISCOPLAS,1`	

Option	Meaning	Type
VALUE	0= use scaling-up scheme, 1=use overstress formula	I

Remarks

1. The strain rate dependent plasticity is normally calculated by scaling up the basic yield stress without strain rate effect. Then the trial stresses are mapped back to the scaled-up yield surface. This algorithm may lead to premature instability. Another technique is to calculate the so-called viscous-plastic strain rate using “overstress” formula. And then the stresses are updated based on this viscous-plastic strain. This technique seems to be more stable then the previous one. For shell elements, this option works when combined with PARAM,SHPLAST,RADIAL. Only DYMAT24 and Johnson-Cook models are supported. For solid elements, this option works only for DYMAT24.
2. For shell elements, PARAM,VISCOPLAS,1, in combination with PARAM,SHPLAST,RADIAL, will use consistent plane stress plasticity algorithm both for strain rate dependent and independent plasticity. This new algorithm is more accurate than 3-D approach.

7 User Defined Services

- Overview 791
- Supported User Defined Subroutines 791
- Build Environment 792
- Implementation with Templates 792
- Implementation without Templates 793
- Building the Service 796
- Environment Variables 796
- Using the Service 796
- Error Handling 797
- Utility Routines 798

Overview

The process of enabling User Defined Services consists of the following four steps:

1. Creating the desired implementation for the User Defined Services in the form of dynamic-link libraries.
2. Defining the location of the user-defined service, service catalogue, and service resource directory.
3. Specifying the proper commands in the model to load the service.
4. Identifying the elements that use the user supplied implementation.

In order to create a dynamic-link library suitable for usage with Dytran, a build environment is delivered to assist users in building the library. Please refer to the *SCA Service Guide* document that describes the features and capabilities of the build system.

Included in the Dytran installation are implementation templates for user subroutines. These templates simplify the implementation of user subroutines, and can be found in the following directory tree:

```
<msc_base>/dytran/<version>/services/{Flow,InitOut,Loads,Materials,Connections}
```

Examples for all UDS services can be found in the following directory tree:

```
<msc_base>/dytran/<version>/UDS-Examples
```

Supported User Defined Subroutines

- Materials
  - EOS - Equation of state model
  - SHR - Shear model
  - YLD - Yield model
  - FAIL - Failure model
  - FAIL1 - Alternative failure model
  - FAIL2 - Second alternative failure model
  - COMP - Failure properties for an orthotropic material for shell element
- Properties
  - ELAS – Elastic Spring
  - SPR - Linear-Elastic Spring
  - VISC - Linear-Damper
- Flow
  - FLOW - Flow definition of an Euler boundary
  - FLOW3 - Alternative flow definition of an Euler Boundary
  - POR - Flow definition of a coupling surface
- InitOut
  - INIT - Initial condition of Lagrangian grid points or Euler elements

- OUT - Output request for Lagrangian grid points or Euler elements

- Loads
  - VEL - Enforced velocity of Lagrangian grid points
  - FUNC - Table definition
- Connections
  - BRK - BJOIN connection failure model

Build Environment

In order to build User Defined Services, the build environment has to be properly configured. Chapter 3 of the *SCA Service Guide* details the process of setting up the build environment. Once the build environment is setup properly, you can easily build service libraries.

Path Descriptions

msc_base: path of the MSC Software base directory.

SCA_OBJECT: path of directory tree containing all temporary files generated by the build. Included are source files, object files, and libraries.

APPS_LOCAL: path of directory tree containing copies of the files generated during a build that will override or add to the files in the SCA application system tree.

The SCA_OBJECT and APPS_LOCAL paths are chosen by the user and can optionally be set by creating or modifying a SCons configuration file in the home directory: SConopts.user. If this file exists with values for SCA_OBJECT and APPS_LOCAL defined, they do not have to be specified in the build command. For the APPS2_SYSTEM variable, <msc_base> has to be replaced with the correct MSC base path prior to building.

Sample SConopts.user file with undefined paths:

```
BUILDTYPE    = 'opt'
APPS_DIR     = 'SCAKernel'
TOOLS_DIR    = 'Tools'
PRODUCT_LINE = 'dytran'

if (SCASCons.MACHINE).startswith('WIN'):
    SCA_OBJECT = '<SCA_OBJECT_PATH>'
    APPS_LOCAL = '<APPS_LOCAL_PATH>'
else:
    SCA_OBJECT = '<SCA_OBJECT_PATH>'
    APPS_LOCAL = '<APPS_LOCAL_PATH>'
```

Implementation with Templates

The user subroutines can be implemented in either C++ or FORTRAN. Calls to the FORTRAN subroutines originate from their respective C++ implementation, so modifying the C++ function body is not recommended if that particular subroutine will be implemented in FORTRAN.

Source files in C++ may also contain various user subroutines, whereas FORTRAN source files have only a single version. Source files containing various user subroutines are either of type Flow, InitOut, Loads, Materials, or Connections.

- Flow.cpp (flow, flow3, por)
- InitOut.cpp (init, out)
- Loads.cpp (vel, func)
- Materials.cpp (fail, fail1, fail2, eos, shr, yld, comp)
- Connections.cpp (brk)

To begin the implementation process, copy the `<msc_base>/dytran/<version>/services/directory` to a directory

with write access; then modify the desired user subroutine in either C++ or FORTRAN.

C++ implementations:

- `Flow/Flow.cpp`
- `InitOut/InitOut.cpp`
- `Loads/Loads.cpp`
- `Materials/Materials.cpp`
- `Properties/Properties.cpp`
- `Connections/Connections.cpp`

FORTRAN implementations

- `Flow/ext_flow3.F`
- `Flow/ext_flow.F`
- `Flow/ext_por.F`
- `InitOut/ext_init.F`
- `InitOut/ext_out.F`
- `Loads/ext_func.F`
- `Loads/ext_vel.F`
- `Materials/ext_comp.F`
- `Materials/ext_fail1.F`
- `Materials/ext_fail2.F`
- `Materials/ext_fail.F`
- `Materials/ext_mateos.F`
- `Materials/ext_matshr.F`
- `Materials/ext_matyld.F`
- `Properties/ext_propelas.F`
- `Properties/ext_propspr.F`
- `Properties/ext_propvisc.F`
- `Connections/ext_brk.F`

Implementation without Templates

Without using the provided templates in the SDK, the first step in implementing a user-defined subroutine is to create a service definition language (SDL) file and component definition language (CDL) file. These two files correspond to one or more interface definition language (IDL) files. The IDL files for user subroutines are provided in the installation.

The next step is to generate the skeleton file for the service. The build environment provides a tool, namely “genskeleton”, to generate the desired skeleton file from an IDL, SDL, and CDL file. Once the skeleton file is generated, you can add the desired functionality to the appropriate methods within the service. By default,

the generated skeleton files will be a C++ source and header file. For further information on how to invoke the genskeleton program, please refer to *SCA Service Guide*.

For illustration purposes, let's assume that you have decided to create a service within "Test" name space for the EXFLOW user subroutine, and the intention is to only implement the interface that is provided by its IDL.

The following code example illustrates the content of the *sdl* file:

```
#ifndef TEST_SDL_INCLUDED
#define TEST_SDL_INCLUDED
#include "SCA/MDSolver/Obj/Uds/Dytran/Flow/Flow.idl"
module Test {
service Test.Flow
{
Interface SCA::MDSolver::Obj::Uds::Dytran::Flow::SCAIMDSolver700Flow;
};
}; // Test
#endif
```

Similarly, the CDL file contains the definition of service (or services), which will form the component. An example for a CDL file is illustrated below:

```
#ifndef TEST_FLOW_CDL_INCLUDED
#define TEST_FLOW_CDL_INCLUDED
#include "test.sdl"
component myTest
{
service Test.Flow;
};
#endif
```

Upon generating the skeleton file with the above, the C++ source file will contain three methods:

- Constructor (a method which is automatically invoked when the service is created)
- Destructor (a method which is automatically invoked when the service is destroyed)
- `runFlow` (the functional method)

The constructor method is called each time the service is initiated. Similarly, the destructor method is called each time the service is terminated. The `runFlow` method is the functional method in which you can add your specific implementation.

The `runFlow` method implements the static signature defined by the Dytran application and will be called by the Dytran application while processing the Eulerian flow boundaries (FLOWX) entries of the input file. Using the parameters passed to `runFlow` method, you will have access to all arguments that are set through the FLOWX entry at input. You can implement your formulation and update the following parameters, returning back to Dytran:

```
DynReal64& pfac,
DynReal64& uxfac,
DynReal64& uyfac,
DynReal64& uzfac,
DynReal64& rhofac,
DynReal64& siefac.
```

More information on these parameters is given in the following sections.

```
#include "Flow.h"
namespace Test {
// Constructor
Flow::Flow(SCAIFlowFactoryAccess* factoryAccess)
                : FlowBase(factoryAccess){}
// Destructor
Flow::~Flow()
{}
SCAResult Flow::runFlow(const SCAInt64 id, const SCAInt64 lenvec, const
DynInt64& iusrzn, const DynReal64& pzon, const DynReal64& qzon, const
DynReal64& uxzon, const DynReal64& uyzon, const DynReal64& uzzon, const
DynReal64& rhozon, const DynReal64& siezon, DynReal64& pfac, DynReal64&
uxfac, DynReal64& uyfac, DynReal64& uzfac, DynReal64& rhofac, DynReal64&
siefac, const DynReal64& sx, const DynReal64& sy, const DynReal64& sz, const
DynReal64& xminel, const DynReal64& xmaxel, const DynReal64& yminel, const
DynReal64& ymaxel, const DynReal64& zminel, const DynReal64& zmaxel,
DynInt64& imatno, SCAInt64& iflwtp, const SCAInt64 ibegin, const SCAInt64
iend)
{
return SCASuccess;
}
}
```

In which:

```
c     id           id of the boundary. is given by the value of "lid"
c                  on the flowex entry
c     lenvec       length of arrays
c     iusrzn       user numbers of elements
c     pzon         pressure in the element
c     qzon         artifical viscosity of the element
c     uxzon        x-velocity of element
c     uyzon        y-velocity of element
c     uzzon        z-velocity of element
c     rhozon       density of elememt
c     siezon       specific internal energy element
c     pfac         pressure at boundary face
c     uxfac        x-velocity at boundary face
c     uyfac        y-velocity at boundary face
c     uzfac        z-velocity at boundary face
c     rhofac       density of inflowing material
c     sx           x-component of the face normal
c     sy           y-component of the face normal
c     sz           z-component of the face normal
c     siefac       specific internal energy of inflowing material
c     xninel       minimal x-coordinate of the boundary face
c     xmaxel       maximal x-coordinate of the boundary face
c     yminel       minimal y-coordinate of the boundary face
c     ymaxel       maximal y-coordinate of the boundary face
c     zminel       minimal z-coordinate of the boundary face
c     zmaxel       maximal z-coordinate of the boundary face
c     imatno       user number of the inflowing material
c     iflwtp       flow type switch:
c                      0  : both in and out flow is allowed
c                      1  : only outflow is allowed
c                      2  : only inflow is allowed
c     ibegin       begin element counter
c     iend         end element counter
```

Building the Service

To build the service, execute the “scons” command at the top level of the source tree. The build environment compiles and links all necessary files, including the modifications to any user subroutine within the source tree. A dynamically loaded library is created and will later be used by the Dytran application at runtime.

If the SConopts.user file does not exist in the home directory or if the paths are not defined in that file, the SCA_OBJECT, APPS_LOCAL, and APPS2_SYSTEM paths must be specified when running scons (provided by the MSC Software Development Kit).

To build the service with SCons, enter the following command:

```
<msc_base>/msc_sdk<prod_ver>/Tools/scons SCA_OBJECT=<SCA_OBJECT_PATH>
APPS_LOCAL=<APPS_LOCAL_PATH> APPS2_SYSTEM=<msc_base>/dytran/<version>/sdk
INTEL_COMP_ROOT=<INTEL_COMPILER_PATH> STRICT_CHECK=False|True
```

Environment Variables

In order to load the generated User Defined Service, within the launch environment, three new environment variables have to be set:

- SCA_LIBRARY_PATH – points to the location where the libraries are to be found.
- SCA_SERVICE_CATALOG – points to the location of the SCA service catalog.
- SCA_RESOURCE_DIR – points to the location of SCA resource directory.

Presence of these three environment variables ensures proper loading of user defined services. For the example provided earlier in this document, these three environment variables should be set as following:

Windows (64 bit):

```
set SCA_LIBRARY_PATH=<APPS_LOCAL>\WIN8664\lib
set SCA_SERVICE_CATALOG=<APPS_LOCAL>\res\SCAServiceCatalog.xml
set SCA_RESOURCE_DIR=<APPS_LOCAL>\res
```

Linux (64 bit):

```
set SCA_LIBRARY_PATH=<APPS_LOCAL>/LX8664/lib
set SCA_SERVICE_CATALOG=<APPS_LOCAL>/res/SCAServiceCatalog.xml
set SCA_RESOURCE_DIR=<APPS_LOCAL>/res
```

Using the Service

In this section, we will describe the modifications to the input file required for using the user-defined service.

In order to identify the service, you will have to create a connection between the service name and a service identifier. This is done through the connect statement in the FMS statement. The syntax for new connect entry is as follows:

```
connect service <group_name> '<service_name>'
```

Where group_name is a name tag (8 characters long) which will be used to identify the service implementation to be used for a specific bulk data entry, and service_name is the name of service, for example:

```
connect service exfl 'SCA.MDSolver.Obj.Uds.Flow'
```

creates a group_name service identifier "exfl", which points to the "SCA.MDSolver.Util.Uds" service.

Available service names for the user subroutines (based on the example implementations) include

- SCA.MDSolver.Obj.Uds.Dytran.Flow (FLOW, POR)
- SCA.MDSolver.Obj.Uds.Dytran.InitOut (INIT, OUT)
- SCA.MDSolver.Obj.Uds.Dytran.Materials (MATYLD, MATFAIL, MATEOS, MATSHR, COMP)
- SCA.MDSolver.Obj.Uds.Dytran.Properties (PROPELAS, PROPSPR, PROPVISC)
- SCA.MDSolver.Obj.Uds.Dytran.Loads (VEL, TABLE)
- SCA.MDSolver.Obj.Uds.Dytran.Connections (EXBRK)

To trigger the service, an appropriate bulk data entry has to be associated with the group name while also specifying the user subroutine in the user subroutine name field of the entry.

Currently available cards for each category of user subroutines comprise:

- FLOWEX/FLOWXSQ/FLOWXDR – Euler flow boundary user subroutines
- POREX – Coupling boundary user subroutines
- GPEXOUT/ELEXOUT – Grid point and element output user subroutines
- TICGEX/TICEEX – Lagrange grid point and Euler element initialization user subroutines
- FAILEX/FAILEX1/FAILEX2 – Material failure model user subroutines
- YLDEX – Material yield model user subroutines
- SHREX – Material shear model runtime info user subroutines
- EOSEX – Material equation of state model runtime info user subroutines
- FORCEEX – Grid point enforced velocity user subroutines
- TABLEEX – Table input user subroutines
- MAT8A with FT=USER1 – Composite material user subroutine
- PELASEX/PSPREX/PVISCEX – Spring/Damper user subroutines

Error Handling

The current implementation provides a limited means for error handling. Upon return from a user defined service method, the value of the return call will be examined. SCASuccess is a constant equal to zero, and if that is returned, the Dytran application assumes that the execution of the User Defined Service has been successful. Anything other than zero is viewed as an error. In the FORTRAN user subroutine implementations, the error code can be set through the error_code variable. Dytran upon return to the main application evaluates this return code and terminates the application if its value is not zero. A message including the nonzero return value is printed in the output file (OUT).

Utility Routines

GET_TIME Routine

Mechanism to get Current Time

Description

The `get_time` routine provides a mechanism to get the current time from the Dytran simulation.

Example

C++:

```
SCAReal64 time;
SCA::MDSolver::Util::DytranUtils::getTimeStep(time);
```

Fortran:

```
real(8) :: time
call GET_TIMESTEP(time)
```

GET_TIMESTEP Routine

Mechanism to get Current Timestep

Description

The `get_timestep` routine provides a mechanism to get the current time step from the Dytran simulation.

Example

C++:

```
SCAReal64 dlth;
SCA::MDSolver::Util::DytranUtils::getTimeStep(dlth);
```

Fortran:

```
real(8) :: time_step
call GET_TIMESTEP(time_step)
```

IGET_STEP Routine

Mechanism to get Current Step Number

Description

The `IGET_STEP` routine provides a mechanism to get the current step number from the Dytran simulation.

Example

C++:

```
SCAInt64 ncycle;
SCA::MDSolver::Util::DytranUtils::getStep(ncycle);
```

Fortran:

```
integer(8) :: ncycle
call IGET_STEP(ncycle)
```

IGET_NSWRAP Routine

Mechanism to get Wrapping Condition

Description

The `IGET_NSWRAP` routine provides a mechanism to get the wrapping condition from the Dytran simulation.

Example

C++:

```
SCAInt64 ncycle;
SCA::MDSolver::Util::DytranUtils::getNsWrap(nswrap);
```

Fortran:

```
integer(8) :: nswrap
call IGET_NSWRAP (nswrap)
```

The following wrapping conditions are possible:

`nswrap = 1`	the cycle limit has been reached
`nswrap = 2`	the time limit has been reached
`nswrap = 3`	the time step has become too small
`nswrap = 5`	cpu time limit exceeded
`nswrap = 8`	an error of unknown fatality occurred
`nswrap = 9`	an error occurred
`nswrap = 13`	a catastrophic error has occurred
`nswrap = 14`	a fatal error has occurred
`nswrap = 16`	an error has occurred in another UDS service

IPUT_NSWRAP Routine

Mechanism to put Wrapping Condition

Description

The `IPUT_NSWRAP` routine provides a mechanism to put a wrapping condition to the Dytran simulation.

Example

C++:

```
SCAInt64 ncycle;
SCA::MDSolver::Util::DytranUtils::putNsWrap(nswrap);
```

Fortran:

```
integer(8) :: nswrap
call IPUT_NSWRAP(nswrap)
```

The only wrapping condition that can be send by `iput_nswrap` is `nswrap=16`.

Flow User-defined Material Flow of an Euler Boundary or Coupling Surface in a Dytran Simulation

Description

The Flow user subroutine is provided to allow the user to define a flow condition that cannot be defined using the existing flow conditions in Dytran. There are three types of Eulerian flows that can be used by the user:

1. EXT_FLOW: Simple flow definition on the boundary of Euler elements. FLOWEX or FLOWXDR or FLOWXSQ with option `MULTI=OFF` must be used.
1. EXT_FLOW3: Advanced flow definition on the boundary of Euler elements. FLOWEX or FLOWXDR or FLOWXSQ with option `MULTI=ON` must be used.
2. EXT_POR: Flow definition on a coupling surface. POREX must be used and be referenced from a COUPOR entry.

Format

IDL

```
module SCA {
module MDSolver {
module Obj {
module Uds {
module Dytran {
module Flow {

typedef SCAInt64 DynInt64[];
typedef SCAReal64 DynReal64[];

interface SCAIMDSolver700Flow : SCAIService
{

    SCAResult usrFlow(     in SCAInt64 id,
                           in SCAInt64 lenvec,
                           in DynInt64 iusrzn,
                           in DynReal64 pzon,
                           in DynReal64 qzon,
                           in DynReal64 uxzon,
                           in DynReal64 uyzon,
                           in DynReal64 uzzon,
                           in DynReal64 rhozon,
                           in DynReal64 siezon,
                           inout DynReal64 pfac,
                           inout DynReal64 uxfac,
                           inout DynReal64 uyfac,
                           inout DynReal64 uzfac,
                           inout DynReal64 rhofac,
                           inout DynReal64 siefac,
                           in DynReal64 sx,
                           in DynReal64 sy,
                           in DynReal64 sz,
```

```
                                in DynReal64 xminel,
                                in DynReal64 xmaxel,
                                in DynReal64 yminel,
                                in DynReal64 ymaxel,
                                in DynReal64 zminel,
                                in DynReal64 zmaxel,
                                inout DynInt64 imatno,
                                inout SCAInt64 iflwtp,
                                in SCAInt64 ibegin,
                                in SCAInt64 iend);


SCAResult usrFlow3(in  SCAInt64 id,
                                in SCAInt64 lenvec,
                                in DynInt64 iusrzn,
                                in DynReal64 pzon,
                                in DynReal64 qzon,
                                in DynReal64 uxzon,
                                in DynReal64 uyzon,
                                in DynReal64 uzzon,
                                in DynReal64 rhozon,
                                in DynReal64 siezon,
                                in DynReal64 fmatzon,
                                inout DynReal64 pfac,
                                inout DynReal64 uxfac,
                                inout DynReal64 uyfac,
                                inout DynReal64 uzfac,
                                inout DynReal64 rhofac,
                                inout DynReal64 siefac,
                                inout DynReal64 fmatfac,
                                in DynReal64 sx,
                                in DynReal64 sy,
                                in DynReal64 sz,
                                in DynReal64 xminf,
                                in DynReal64 xmaxf,
                                in DynReal64 yminf,
                                in DynReal64 ymaxf,
                                in DynReal64 zminf,
                                in DynReal64 zmaxf,
                                inout DynInt64 imateul,
                                in SCAInt64 lmmult,
                                inout SCAInt64 iflwtp,
                                inout SCAInt64 nmateul,
                                in SCAInt64 ibegin,
                                in SCAInt64 iend);


SCAResult usrPor(       in SCAInt64 id,
                                in SCAInt64 lenvec,
                                in SCAInt64 ibegin,
                                in SCAInt64 iend,
                                in DynInt64 ieul,
                                in DynReal64 press,
                                in DynReal64 densty,
```

```
                                in DynReal64 sie,
                                in DynReal64 xvel,
                                in DynReal64 yvel,
                                in DynReal64 zvel,
                                in DynReal64 ux,
                                in DynReal64 uy,
                                in DynReal64 uz,
                                in DynReal64 area,
                                in DynReal64 sx,
                                in DynReal64 sy,
                                in DynReal64 sz,
                                in DynReal64 visc,
                                inout DynReal64 vtrans,
                                inout DynReal64 dmass,
                                inout DynReal64 fpress,
                                inout DynReal64 fsie,
                                inout DynReal64 xvelf,
                                inout DynReal64 yvelf,
                                inout DynReal64 zvelf,
                                inout DynReal64 coeffv,
                                inout DynInt64 imatno);

};

}; //Flow
}; //Dytran
}; //Uds
}; //Obj
}; //MDSolver
}; //SCA
```

C++

```
SCAResult Flow::usrFlow(const SCAInt64 id, const SCAInt64 lenvec, const DynInt64& iusrzn, const DynReal64& pzon,
const DynReal64& qzon, const DynReal64& uxzon, const DynReal64& uyzon, const DynReal64& uzzon, const DynReal64&
rhozon, const DynReal64& siezon, DynReal64& pfac, DynReal64& uxfac, DynReal64& uyfac, DynReal64& uzfac, DynReal64&
rhofac, DynReal64& siefac, const DynReal64& sx, const DynReal64& sy, const DynReal64& sz, const DynReal64& xminel,
const DynReal64& xmaxel, const DynReal64& yminel, const DynReal64& ymaxel, const DynReal64& zminel, const DynReal64&
zmaxel, DynInt64& imatno, SCAInt64& iflwtp, const SCAInt64 ibegin, const SCAInt64 iend)

{

  SCAInt64 err_code = 0;

  return err_code;

}

SCAResult Flow::usrFlow3(const SCAInt64 id, const SCAInt64 lenvec, const DynInt64& iusrzn, const DynReal64& pzon,
const DynReal64& qzon, const  DynReal64& uxzon, const DynReal64& uyzon, const DynReal64& uzzon, const DynReal64&
rhozon, const DynReal64& siezon, const DynReal64& fmatzon, DynReal64& pfac, DynReal64& uxfac, DynReal64& uyfac,
DynReal64& uzfac, DynReal64& rhofac, DynReal64& siefac, DynReal64& fmatfac, const DynReal64& sx, const DynReal64&
```

```
sy, const DynReal64& sz, const DynReal64& xminf, const DynReal64& xmaxf, const DynReal64& yminf, const DynReal64&
ymaxf, const DynReal64& zminf, const DynReal64& zmaxf, DynInt64& imateul, const SCAInt64 lmmult, SCAInt64& iflwtp,
SCAInt64& nmateul, const SCAInt64 ibegin, const SCAInt64 iend)

{

  SCAInt64 err_code = 0;

  return err_code;

}


SCAResult Flow::usrPor(const SCAInt64 id, const SCAInt64 lenvec, const SCAInt64 ibegin, const SCAInt64 iend, const
DynInt64& ieul, const DynReal64& press, const DynReal64& densty, const DynReal64& sie, const DynReal64& xvel, const
DynReal64& yvel, const DynReal64& zvel, const DynReal64& ux, const DynReal64& uy, const DynReal64& uz, const
DynReal64& area, const DynReal64& sx, const DynReal64& sy, const DynReal64& sz, const DynReal64& visc,  DynReal64&
vtrans,  DynReal64& dmass, DynReal64& fpress, DynReal64& fsie,  DynReal64& xvelf, DynReal64& yvelf,  DynReal64&
zvelf,  DynReal64& coeffv, DynInt64& imatno)

{

  SCAInt64 err_code = 0;

  return err_code;
}
```

FORTRAN

```
      subroutine EXT_FLOW(id,lenvec,iusrzn,pzon,qzon,uxzon,
     +                    uyzon,uzzon,rhozon,siezon,pfac,uxfac,
     +                    uyfac,uzfac,rhofac,siefac,sx,sy,sz,
     +                    xminel,xmaxel,yminel,ymaxel,zminel,zmaxel,
     +                    imatno,iflwtp,ibegin,iend,err_code)
C
C
    implicit none
      integer(8),intent(in) :: id, lenvec, ibegin, iend
      integer(8),intent(inout) :: iflwtp
      integer, intent(inout) :: err_code
C
      real(8), intent(inout),dimension(lenvec):: pzon, qzon
      real(8), intent(inout),dimension(lenvec):: rhozon, siezon
      real(8), intent(inout),dimension(lenvec):: uxzon, uyzon, uzzon
      real(8), intent(inout),dimension(lenvec):: uxfac, uyfac, uzfac
      real(8), intent(inout),dimension(lenvec):: rhofac, siefac, pfac
      integer(8),intent(inout),dimension(lenvec):: imatno
      integer(8),intent(in), dimension(lenvec):: iusrzn
      real(8), intent(in),dimension(lenvec):: sx, sy, sz
      real(8), intent(in),dimension(lenvec):: xminel, xmaxel, yminel
      real(8), intent(in),dimension(lenvec):: ymaxel, zminel, zmaxel
C
      return
      end subroutine
```

id	id of the boundary. Is given by the value of "fid" on the flowex entry.
lenvec	length of arrays.
iusrzn	user numbers of element.
pzon	pressure in the element.
qzon	artificial viscosity of the element.
uxzon	x-velocity of element.
uyzon	y-velocity of element.

uzzon	z-velocity of element.
rhozon	density of element.
siezon	specific internal energy element.
pfac	pressure at boundary face.
uxfac	x-velocity at boundary face.
uyfac	y-velocity at boundary face.
uzfac	z-velocity at boundary face.
rhofac	density of inflowing material.
sx	x-component of the face normal.
sy	y-component of the face normal.
sz	z-component of the face normal.
siefac	specific internal energy of inflowing material.
xminel	minimal x-coordinate of the boundary face.
xmaxel	maximal x-coordinate of the boundary face.
yminel	minimal y-coordinate of the boundary face.
ymaxel	maximal y-coordinate of the boundary face.
zminel	minimal z-coordinate of the boundary face.
zmaxel	maximal z-coordinate of the boundary face.
imatno	user number of the inflowing material.
ifltwp	flow type switch: 0: both in and out flow is allowed 1: only outflow is allowed 2: only inflow is allowed
ibegin	begin element counter.
iend	end element counter.

```
subroutine EXT_FLOW3(id,lenvec,iusrzn,pzon,qzon,uxzon,
     +               uyzon,uzzon,rhozon,siezon,fmatzon,pfac,uxfac,
     +               uyfac,uzfac,rhofac,siefac,fmatfac,sx,sy,sz,
     +               xminf,xmaxf,yminf,ymaxf,zminf,zmaxf,
     +               imateul,lmmult,iflwtp,nmateul,ibegin,iend,
     +               err_code)
c
      implicit none
c
      integer(8),intent(in) :: id, lenvec,  ibegin, iend, lmmult
      integer(8),intent(inout) :: iflwtp,nmateul,err_code
c
      real(8), intent(in),dimension(lenvec):: pzon, qzon, uxzon, uyzon
      real(8), intent(in),dimension(lenvec):: uzzon
      real(8), intent(in),dimension(lenvec,lmmult):: rhozon, siezon
```

```
real(8), intent(in),dimension(lenvec,lmmult):: fmatzon
real(8), intent(in),dimension(lenvec):: xminf,xmaxf,yminf,ymaxf
real(8), intent(in),dimension(lenvec):: zminf,zmaxf,sx,sy,sz
real(8), intent(inout),dimension(lenvec,lmmult):: uxfac, uyfac
real(8), intent(inout),dimension(lenvec,lmmult):: uzfac, fmatfac
real(8), intent(inout),dimension(lenvec,lmmult):: rhofac,siefac
real(8), intent(inout),dimension(lenvec):: pfac
integer(8), intent(inout),dimension(lmmult):: imateul
integer(8), intent(in),dimension(lenvec):: iusrzn

return
end subroutine
```

id	id of the boundary. Is given by the value of "fid" on the flowex entry.
lenvec	length of arrays.
iusrzn	user numbers of elements.
pzon	pressure in the element.
qzon	artificial viscosity of the element
uxzon	x-velocity of element.
uyzon	y-velocity of element.
uzzon	z-velocity of element.
rhozon(,nd)	density of element for Eulerian material nd.
	the user number of Eulerian material nd is imateul(nd).
siezon(,nd)	specific internal energy element for Eulerian material nd.
fmatzon(,nd)	material fraction element for Eulerian material nd.
pfac	pressure at boundary face.
uxfac	x-velocity at boundary face.
uyfac	y-velocity at boundary face.
uzfac	z-velocity at boundary face.
rhofac(,nd)	density of inflowing material.
siefac(,nd)	specific internal energy of inflowing material
fmatfac(,nd)	material fraction of face.
sx	x-component of the face normal.
sy	y-component of the face normal.
sz	z-component of the face normal.
xminf	minimal x-coordinate of the boundary face.
xmaxf	maximal x-coordinate of the boundary face.
yminf	minimal y-coordinate of the boundary face.
ymaxf	maximal y-coordinate of the boundary face.

zminf	minimal z-coordinate of the boundary face.
zmaxf	maximal z-coordinate of the boundary face.
imateul	list of Eulerian materials.
lmmult	maximal number of Euler materials that is allowed.
ifltwp	flow type switch: 0: both in and out flow is allowed 1: only outflow is allowed 2: only inflow is allowed
nmateul	number of eulerian materials.
ibegin	begin element counter.
iend	end element counter.

```
      subroutine EXT_POR(id,lenvec,ibegin,iend,ieul,
     +                   press,densty,sie,xvel,yvel,zvel,ux,uy,uz,
     +                   area,sx,sy,sz,visc,vtrans,dmass,fpress,fsie,
     +                   xvelf,yvelf,zvelf,coeffv,imatno,err_code)
c
      implicit none
       integer(8),intent(in), dimension(*):: ieul,id
       integer(8),intent(inout), dimension(*):: imatno
       integer(8),intent(in) :: lennam, lenvec, ibegin, iend
       integer, intent(inout) :: err_code
c
       real(8), intent(in), dimension(*):: press,densty,sie
       real(8), intent(in), dimension(*):: xvel,yvel,zvel,ux,uy,uz
       real(8), intent(in), dimension(*):: area,sx,sy,sz,visc
       real(8), intent(inout), dimension(*):: vtrans,dmass,fpress
       real(8), intent(inout), dimension(*):: fsie,xvelf,yvelf,zvelf
       real(8), intent(inout), dimension(*):: coeffv

      return
      end subroutine
```

id	id of the porosity definition. is given by the value of "pid" on the porex entry.
lenvec	length of arrays.
ibegin	begin segment counter.
iend	end segment counter.
ieul	user number of euler element connected to the segment.
press	pressure in the element connected to the segment.
density	density in the element connected to the segment.
sie	sie in the element.
xvel	x-velocity component of the element.
yvel	y-velocity component of the element.

zvel	z-velocity component of the element.
ux	x-velocity component of the moving face the segment is part of.
uy	y-velocity component of the moving face.
uz	z-velocity component of the moving face.
area	total segment area.
sx	x-component of the segment normal.
sy	y-component of the segment normal.
sz	z-component of the segment normal.
visc	artificial viscosity of the element connected to a segment.
vtrans	transported volume through the segment.
dmass	transported mass through the segment.
fpress	pressure at the segment face.
fsie	specific internal energy at the segment face.
xvelf	x-velocity component of the fluid at the segment face.
yvelf	y-velocity component of the fluid at the segment face.
zvelf	z-velocity component of the fluid at the segment face.
coeffv	porosity coefficient of the segment.
imatno	user number of the inflowing material.

InitOut User-defined Initialization or Output in a Dytran Simulation

Description

The InitOut user subroutine is provided to allow the user to define an initial condition in a Dytran simulation. `InitOut` contain two types of user subroutines:

1. EXT_INIT: User-defined initialization of elements or grid points at the start of the simulation. TICCEX or TICGEX must be used to activate this user subroutine.
2. EXT_OUT: User-defined output request. ELEXOUT or GPEXOUT must be used to activate this user subroutine.

Format

IDL

```
module SCA {
module MDSolver {
module Obj {
module Uds {
module Dytran {
module InitOut {

typedef SCAInt64 DynInt64[];
typedef SCAReal64 DynReal64[];

interface SCAIMDSolver700InitOut : SCAIService
{

      SCAResult usrOut(    in SCAString cname,
                           in SCAInt64 lennam,
                           in SCAInt64 lenvec,
                           in DynInt64 iusrzn,
                           in SCAInt64 nztype,
                           in SCAInt64 ibegin,
                           in SCAInt64 iend,
                           in SCAInt64 nmiid,
                           in SCAInt64 nmxid,
                           inout DynInt64 ivars,
                           inout DynReal64 xvars,
                           in DynReal64 xminel,
                           in DynReal64 xmaxel,
                           in DynReal64 yminel,
                           in DynReal64 ymaxel,
                           in DynReal64 zminel,
                           in DynReal64 zmaxel,
                           inout DynInt64 iwork,
                           inout DynReal64 xwork);

        SCAResult usrInit( in SCAString cname,
                           in SCAInt64 lennam,
                           in SCAInt64 lenvec,
```

```
                                in DynInt64 ngpel,
                                in SCAInt64 ibegin,
                                in SCAInt64 iend,
                                inout DynReal64 density,
                                inout DynReal64 sie,
                                inout DynReal64 xvel,
                                inout DynReal64 yvel,
                                inout DynReal64 zvel,
                                inout DynInt64 ivoid,
                                in DynInt64 nummat,
                                in DynReal64 xminel,
                                in DynReal64 xmaxel,
                                in DynReal64 yminel,
                                in DynReal64 ymaxel,
                                in DynReal64 zminel,
                                in DynReal64 zmaxel);

};

}; //InitOut
}; //Dytran
}; //Uds
}; //Obj
}; //MDSolver
}; //SCA
```

C++

```
SCAResult InitOut::usrOut(const SCAString cname, const SCAInt64 lennam, const SCAInt64 lenvec, const DynInt64&
iusrzn, const SCAInt64 nztype, const SCAInt64 ibegin, const SCAInt64 iend, const SCAInt64 nmiid, const SCAInt64
nmxid, DynInt64& ivars, DynReal64& xvars, const DynReal64& xminel, const DynReal64& xmaxel, const DynReal64& yminel,
const DynReal64& ymaxel, const DynReal64& zminel, const DynReal64& zmaxel, DynInt64& iwork, DynReal64& xwork) {
return SCASuccess;
}

SCAResult InitOut::usrInit(const SCAString cname, const SCAInt64 lennam, const SCAInt64 lenvec, const DynInt64&
ngpel, const SCAInt64 ibegin, const SCAInt64 iend,  DynReal64& density,  DynReal64& sie, DynReal64& xvel, DynReal64&
yvel, DynReal64& zvel,  DynInt64& ivoid, const DynInt64& nummat , const DynReal64& xminel, const DynReal64&  xmaxel,
const DynReal64& yminel, const DynReal64& ymaxel, const DynReal64& zminel ,const DynReal64& zmaxel) {
return SCASuccess;
}
```

FORTRAN

```
      subroutine EXT_OUT
     + (cname,lennam,lenvec,iusrzn,nztype,
     +  ibegin,iend,nmiid,nmxid,ivars,xvars,
     +  xminel,xmaxel,yminel,ymaxel,zminel,zmaxel,
     +  iwork,xwork,err_code)
c
      implicit none
      character(*) :: cname
      integer(8),intent(inout) :: err_code
      integer*8,intent(in)::lennam,ibegin,iend,lenvec,nmiid
      integer*8,intent(in)::nmxid,nztype
      integer*8,intent(in)::iusrzn(lenvec)
      integer*8,intent(inout)::ivars(nmiid,lenvec),iwork(lenvec)
      real*8,intent(in)::xminel(lenvec),xmaxel(lenvec)
      real*8,intent(in)::yminel(lenvec),ymaxel(lenvec)
      real*8,intent(in)::zminel(lenvec),zmaxel(lenvec)
      real*8,intent(inout)::xvars(nmxid,lenvec)
      real*8,intent(inout)::xwork(lenvec)
```

```
c
      return
      end subroutine
```

cname	name of the output request. Is given by the value of "nid" on the ELEXOUT or GPEXOUT entry.
lennam	length of cname.
lenvec	length of arrays.
iusrzn	user numbers of elements.
nztype	type of element: 2 Beam, rod, sprint and damper 3 Triangular shell 4 Quadratic shell 5 Membrane 6 Dummy triangular 7 Dummy quadratic 8 Langrangian solid 9 Eulerian solid (hydrodynamic) 10 Eulerian solid (with strength) 11 Eulerian solid (multimaterial)
ibegin	start element counter.
iend	end element counter.
nmiid	number of integer variables.
nmxid	number of real variables.
ivars(nv,nz)	integer variable nv for element nz.
	A list of integer element variables is given in the OUT file.
xvars(nv,nz)	real variable nv for element nz.
	A list of real element variables is given in the OUT file.
xminel	minimal x-coordinate of the element.
xmaxel	maximal x-coordinate of the element.
yminel	minimal y-coordinate of the element.
ymaxel	maximal y-coordinate of the element.
zminel	minimal z-coordinate of the element.
zmaxel	maximal z-coordinate of the element.
iwork	integer work memory.
xwork	real work memory.
err_code	returned error code; 0 = success, otherwise fail.

```
      subroutine EXT_INIT(cname,lennam,lenvec,iusrzn,
     +                    ibegin,iend, density, sie, xvel, yvel, zvel,
     +                    ivoid,nummat,
     +                    xminel,xmaxel,yminel,ymaxel,zminel,zmaxel,
     +                    err_code)
c
      implicit none
      character(lennam) :: cname
      integer(8), intent(in) :: lennam, lenvec,ibegin,iend
      integer(8), intent(in), dimension(lenvec):: iusrzn,ivoid,nummat
      integer, intent(inout) :: err_code
      real(8), intent(inout), dimension(lenvec):: density, sie
      real(8), intent(inout), dimension(lenvec):: xvel, yvel, zvel
      real(8), intent(in), dimension(lenvec):: xminel,xmaxel,yminel
      real(8), intent(in), dimension(lenvec):: ymaxel,zminel,zmaxel
c
      return
      end subroutine
```

cname	name of the boundary. Is given by the value of sid on the TICEEX or TICGEX entry.
lennam	length of cname.
lenvec	length of arrays.
iusrzn	user numbers of elements.
ibegin	begin element counter.
iend	end element counter.
density	density of element.
sie	specific internal energy element.
uxvel	x-velocity of element/ gridpoint.
uyvel	y-velocity of element/ gridpoint.
uzvel	z-velocity of element/ gridpoint.
ivoid	void flag of element.
nummat	number of materials in element.
xminel	minimal x-coordinate of element/ gridpoint.
xmaxel	maximal x-coordinate of element/ gridpoint.
yminel	minimal y-coordinate of element/ gridpoint.
ymaxel	maximal y-coordinate of element/ gridpoint.
zminel	minimal z-coordinate of element/ gridpoint.
zmaxel	maximal z-coordinate of element/ gridpoint.
err_code	is the returned error code; 0 = success, otherwise fail.

Materials

User-defined Material Definition in a Dytran Simulation

Description

The Materials user subroutine is provided to allow the user to define an Eulerian material model in a Dytran simulation. Materials permits the user definition of an equation of state, yield, shear and two failure models by user subroutines:

1. EXT_MATYLD: User defined yield material model using YLDEX.
2. EXT_MATEOS: User defined equation of state material model using EOSEX.
3. EXT_MATFAIL: Simplified user defined failure material model using FAILEX.
4. EXT_MATFAIL1: Alternative user defined failure material model using FAILEX1.
5. EXT_MATFAIL2: Advanced user defined failure material model using FAILEX2.
6. EXT_SHR: User defined shear material model using SHREX.
7. EXT_COMP: User defined composite material model. MAT8A with option `FT=USER1` must be used.

Format

IDL

```
module SCA {
module MDSolver {
module Obj {
module Uds {
module Dytran {
module Materials {

typedef SCAInt64 DynInt64[];
typedef SCAReal64 DynReal64[];

interface SCAIMDSolver700Materials : SCAIService
{
SCAResult usrExcomp(in SCAInt64 id,
                    in SCAInt64 lenvec,
                    in SCAInt64 ishrpr,
                    in SCAReal64 ymx,
                    in SCAReal64 ymy,
                    in SCAReal64 xnuy,
                    in SCAReal64 sxy,
                    in SCAReal64 syz,
                    in SCAReal64 szx,
                    in SCAReal64 xt,
                    in SCAReal64 xc,
                    in SCAReal64 yt,
                    in SCAReal64 yc,
                    in SCAReal64 sc,
                    in SCAReal64 capa,
                    in DynReal64 cm,
                    in SCAInt64 ibegin,
```

```
                        in SCAInt64 iend,
                        in SCAInt64 nadvar,
                        in SCAInt64 isubly,
                        inout DynReal64 sig1,
                        inout DynReal64 sig2,
                        inout DynReal64 sig4,
                        inout DynReal64 sig5,
                        inout DynReal64 sig6,
                        inout DynReal64 sig51,
                        inout DynReal64 sig62,
                        in DynReal64 d1,
                        in DynReal64 d2,
                        in DynReal64 d3,
                        in DynReal64 d4,
                        in DynReal64 d5,
                        in DynReal64 d6,
                        in DynReal64 d7,
                        in DynReal64 d8,
                        in DynReal64 dout1,
                        in DynReal64 dout2,
                        in DynReal64 dout4,
                        inout DynReal64 eft,
                        inout DynReal64 efc,
                        inout DynReal64 esf,
                        inout DynReal64 emt,
                        inout DynReal64 emc,
                        in DynReal64 q1,
                        in DynReal64 q2,
                        inout DynReal64 einc,
                        in DynReal64 fcap,
                        inout DynReal64 fail,
                        inout DynReal64 falnm1,
                        inout DynReal64 efail,
                        inout DynReal64 fail2,
                        inout DynReal64 user,
                        inout DynReal64 efib,
                        inout DynReal64 emtx,
                        inout DynReal64 eshr,
                        in SCAInt64 lmcmar
);

SCAResult Materials::usrExcomp(const SCAInt64 id,
                               const SCAInt64 lenvec,
                               const SCAInt64 ishrpr,
                               const SCAReal64 ymx,
                               const SCAReal64 ymy,
                               const SCAReal64 xnuy,
                               const SCAReal64 sxy,
                               const SCAReal64 syz,
                               const SCAReal64 szx,
                               const SCAReal64 xt,
                               const SCAReal64 xc,
                               const SCAReal64 yt,
                               const SCAReal64 yc,
```

```
                                    const SCAReal64 sc,
                                    const SCAReal64 capa,
                                    const DynReal64& cm,
                                    const SCAInt64 ibegin,
                                    const SCAInt64 iend,
                                    const SCAInt64 nadvar,
                                    const SCAInt64 isubly,
                                    DynReal64& sig1,
                                    DynReal64& sig2,
                                    DynReal64& sig4,
                                    DynReal64& sig5,
                                    DynReal64& sig6,
                                    DynReal64& sig51,
                                    DynReal64& sig62,
                                    const DynReal64& d1,
                                    const DynReal64& d2, const
                                    DynReal64& d3,
                                    const DynReal64& d4,
                                    const DynReal64& d5,
                                    const DynReal64& d6,
                                    const DynReal64& d7,
                                    const DynReal64& d8,
                                    const DynReal64& dout1,
                                    const DynReal64& dout2,
                                    const DynReal64& dout4,
                                    DynReal64& eft,
                                    DynReal64& efc,
                                    DynReal64& esf,
                                    DynReal64& emt,
                                    DynReal64& emc,
                                    const DynReal64& q1,
                                    const DynReal64& q2,
                                    DynReal64& einc, const
                                    DynReal64& fcap,
                                    DynReal64& fail,
                                    DynReal64& falnm1,
                                    DynReal64& efail,
                                    DynReal64& fail2,
                                    DynReal64& user,
                                    DynReal64& efib,
                                    DynReal64& emtx,
                                    DynReal64& eshr,
                                    const SCAInt64 lmcmar
)

     SCAResult usrMatyld(in SCAInt64 id,
                         in SCAInt64 lenvec,
                         in DynInt64 iusrzn,
                         in DynReal64 pres,
                         in DynReal64 edis,
                         in DynReal64 sie,
                         in DynReal64 rho,
                         in DynReal64 fburn,
                         inout DynReal64 effpls,
```

```
                                in DynReal64 zmass,
                                in DynReal64 effsts,
                                in DynReal64 twoj2,
                                in DynReal64 effsr,
                                inout DynReal64 usrvr1,
                                inout DynReal64 usrvr2,
                                in DynReal64 relv,
                                in DynReal64 sxxo,
                                in DynReal64 syyo,
                                in DynReal64 szzo,
                                in DynReal64 sxyo,
                                in DynReal64 syzo,
                                in DynReal64 sxzo,
                                in DynReal64 sxxt,
                                in DynReal64 syyt,
                                in DynReal64 szzt,
                                in DynReal64 sxyt,
                                in DynReal64 syzt,
                                in DynReal64 sxzt,
                                in DynReal64 dexx,
                                in DynReal64 deyy,
                                in DynReal64 dezz,
                                in DynReal64 dexy,
                                in DynReal64 deyz,
                                in DynReal64 dezx,
                                in DynReal64 tdet,
                                inout DynReal64 yldsq,
                                inout DynReal64 damage,
                                inout DynReal64 softe,
                                inout DynReal64 volpls,
                                in SCAInt64 ibegin,
                                in SCAInt64 iend
        );

         SCAResult usrMateos(in SCAInt64 id,
                                in SCAInt64 lenvec,
                                in SCAInt64 ibegin,
                                in SCAInt64 iend,
                                in DynReal64 rho,
                                in DynReal64 dv,
                                in DynReal64 devis,
                                in DynReal64 xmass,
                                in DynReal64 fburn,
                                in DynReal64 pold,
                                in DynReal64 sieold,
                                inout DynReal64 pnew,
                                inout DynReal64 sienew,
                                inout DynReal64 clnew,
                                inout DynReal64 grungm,
                                inout DynReal64 damage,
                                inout DynReal64 softe,
                                inout DynReal64 volpls
    );
```

```
SCAResult usrMatfail(in SCAInt64 id,
                     in SCAInt64 lenvec,
                     in SCAInt64 ibegin,
                     in SCAInt64 iend,
                     in DynReal64 eplas,
                     in DynReal64 effsts,
                     in DynReal64 pres,
                     in DynReal64 sie,
                     in DynReal64 rho,
                     inout DynReal64 ffail,
                     in DynInt64 iusrzn

);

SCAResult usrMatfail1(in SCAInt64 id,
                      in SCAInt64 lenvec,
                      in SCAInt64 ibegin,
                      in SCAInt64 iend,
                      in DynInt64 iusrzn,
                      in DynReal64 txx,
                      in DynReal64 tyy,
                      in DynReal64 tzz,
                      in DynReal64 txy,
                      in DynReal64 tyz,
                      in DynReal64 txz,
                      in DynReal64 depxx,
                      in DynReal64 depyy,
                      in DynReal64 depzz,
                      in DynReal64 depxy,
                      in DynReal64 depyz,
                      in DynReal64 depxz,
                      in DynReal64 epxx,
                      in DynReal64 epyy,
                      in DynReal64 epzz,
                      in DynReal64 epxy,
                      in DynReal64 epyz,
                      in DynReal64 epxz,
                      in DynReal64 exx,
                      in DynReal64 eyy,
                      in DynReal64 ezz,
                      in DynReal64 exy,
                      in DynReal64 eyz,
                      in DynReal64 exz,
                      in DynReal64 usrvr1,
                      in DynReal64 usrvr2,
                      inout DynReal64 tstepzn,
                      inout DynReal64 ffail);


SCAResult usrMatfail2(in SCAInt64 id,
                      in SCAInt64 lenvec,
                      in SCAInt64 ibegin,
                      in SCAInt64 iend,
```

```
                        in DynReal64 eplas,
                        in DynReal64 effsts,
                        in DynReal64 pres,
                        in DynReal64 sie,
                        in DynReal64 rho,
                        inout DynReal64 damage,
                        inout DynReal64 softe,
                        inout DynReal64 volpls,
                        in DynReal64 depi,
                        inout DynReal64 sxx,
                        inout DynReal64 syy,
                        inout DynReal64 szz,
                        inout DynReal64 sxy,
                        inout DynReal64 syz,
                        inout DynReal64 sxz,
                        in DynReal64 sxxo,
                        in DynReal64 syyo,
                        in DynReal64 szzo,
                        in DynReal64 sxyo,
                        in DynReal64 syzo,
                        in DynReal64 sxzo,
                        in DynReal64 shear
);

     SCAResult usrMatshr(in SCAInt64 id,
                         in SCAInt64 lenvec,
                         in DynReal64 exx,
                         in DynReal64 eyy,
                         in DynReal64 ezz,
                         in DynReal64 exy,
                         in DynReal64 eyz,
                         in DynReal64 ezx,
                         in DynReal64 pres,
                         in DynReal64 edis,
                         in DynReal64 sie,
                         in DynReal64 rho,
                         in DynReal64 fburn,
                         in DynReal64 zmass,
                         inout DynReal64 shear,
                         in SCAInt64 begin,
                         in SCAInt64 iend);

};

}; //Materials
}; //Dytran
}; //Uds
}; //Obj
}; //MDSolver
}; //SCA
```

C++

```
SCAResult Materials::usrMatyld(const SCAInt64 id, const SCAInt64 lenvec, const DynInt64& iusrzn, const DynReal64&
pres, const DynReal64& edis, const DynReal64& sie, const DynReal64& rho, const DynReal64& fburn, DynReal64& effpls,
const DynReal64& zmass, const DynReal64& effsts, const DynReal64& twoj2, const DynReal64& effsr, DynReal64& usrvr1,
DynReal64& usrvr2, const DynReal64& relv, const DynReal64& sxxo, const DynReal64& syyo, const DynReal64& szzo, const
```

```
DynReal64& sxyo, const DynReal64& syzo, const DynReal64& sxzo, const DynReal64& sxxt, const DynReal64& syyt, const
DynReal64& szzt, const DynReal64& sxyt, const DynReal64& syzt, const DynReal64& sxzt, const DynReal64& dexx, const
DynReal64& deyy, const DynReal64& dezz, const DynReal64& dexy, const DynReal64& deyz, const DynReal64& dezx, const
DynReal64& tdet, DynReal64& yldsq, DynReal64& damage, DynReal64& softe, DynReal64& volpls, const SCAInt64 ibegin,
const SCAInt64 iend)
SCAResult EulMat::usrMatyld(const SCAString flname, const SCAInt64 lennam, const SCAInt64 lenvec, const DynInt64&
iusrzn, const DynReal64& pres, const DynReal64& edis, const DynReal64& sie, const DynReal64& rho, const DynReal64&
fburn, DynReal64& effpls, const DynReal64& zmass, const DynReal64& effsts, const DynReal64& twoj2, const DynReal64&
effsr, DynReal64& usrvr1, DynReal64& usrvr2, const DynReal64& relv, const DynReal64& sxxo, const DynReal64& syyo,
const DynReal64& szzo, const DynReal64& sxyo, const DynReal64& syzo, const DynReal64& sxzo, const DynReal64& sxxt,
const DynReal64& syyt, const DynReal64& szzt, const DynReal64& sxyt, const DynReal64& syzt, const DynReal64& sxzt,
const DynReal64& dexx, const DynReal64& deyy, const DynReal64& dezz, const DynReal64& dexy, const DynReal64& deyz,
const DynReal64& dezx, const DynReal64& tdet, DynReal64& yldsq, DynReal64& damt, DynReal64& softe, DynReal64&
volpls, const SCAInt64 lst1, const SCAInt64 lfin1)
{
return SCASuccess;
}

SCAResult Materials::usrMateos(const SCAInt64 id, const SCAInt64 lenvec, const SCAInt64 ibegin, const SCAInt64 iend,
const DynReal64& rho, const DynReal64& dv, const DynReal64& devis, const DynReal64& xmass, const DynReal64& fburn,
const DynReal64& pold, const DynReal64& sieold, DynReal64& pnew, DynReal64& sienew, DynReal64& clnew, DynReal64&
grungm, DynReal64& damage, DynReal64& softe, DynReal64& volpls)
SCAResult EulMat::usrMateos(const SCAString flname, const SCAInt64 lennam, const SCAInt64 lenvec, const SCAInt64
ibegin, const
SCAInt64 iend, const DynReal64& rho, const DynReal64& dv, const DynReal64& devis, const DynReal64& xmass, const
DynReal64& fburn,
const DynReal64& pold, const DynReal64& sieold, DynReal64& pnew, DynReal64& sienew, DynReal64& clnew, DynReal64&
grungm, DynReal64&
damg, DynReal64& softe, DynReal64& vlpl)
{
return SCASuccess;
}

SCAResult Materials::usrMatfail(const SCAInt64 id, const SCAInt64 lenvec, const SCAInt64 ibegin, const SCAInt64
iend, const DynReal64& eplas, const DynReal64& effsts, const DynReal64& pres, const DynReal64& sie, const DynReal64&
rho, DynReal64& ffail, const DynInt64& iusrzn)
SCAResult EulMat::usrMatfail(const SCAString flname, const SCAInt64 lennam, const SCAInt64 lenvec, const SCAInt64
lst1, const SCAInt64 lfin1, const DynReal64& eplas, const DynReal64& effsts, const DynReal64& pres, const DynReal64&
sie, const DynReal64& rho, DynReal64& ffail, const DynInt64& iusrzn)
{
return SCASuccess;
}

SCAResult Materials::usrMatfail1(const SCAInt64 id, const SCAInt64 lenvec, const SCAInt64 ibegin, const SCAInt64
iend, const DynInt64& iusrzn, const DynReal64& txx, const DynReal64& tyy, const DynReal64& tzz, const DynReal64&
txy, const DynReal64& tyz, const DynReal64& txz, const DynReal64& depxx, const DynReal64& depyy, const DynReal64&
depzz, const DynReal64& depxy, const DynReal64& depyz, const DynReal64& depxz, const DynReal64& epxx, const
DynReal64& epyy, const DynReal64& epzz, const DynReal64& epxy, const DynReal64& epyz, const DynReal64& epxz, const
DynReal64& exx, const DynReal64& eyy, const DynReal64& ezz, const DynReal64& exy, const DynReal64& eyz, const
DynReal64& exz, const DynReal64& usrvr1, const DynReal64& usrvr2, DynReal64& tstepzn, DynReal64& ffail )
{
return SCASuccess;
}

SCAResult Materials::usrMatfail2(const SCAInt64 id, const SCAInt64 lenvec, const SCAInt64 ibegin, const SCAInt64
iend, const DynReal64& eplas,const DynReal64& effsts, const DynReal64& pres,const DynReal64& sie,const DynReal64&
rho, DynReal64& damage, DynReal64& softe, DynReal64& volpls, const DynReal64& depi, DynReal64& sxx, DynReal64& syy,
DynReal64& szz, DynReal64& sxy, DynReal64& syz, DynReal64& sxz, const DynReal64& sxxo, const DynReal64& syyo, const
DynReal64& szzo, const DynReal64& sxyo, const DynReal64& syzo, const DynReal64& sxzo, const DynReal64&
shear)SCAResult EulMat::usrMatfail2(const SCAString flname, const SCAInt64 lennam, const SCAInt64 lenvec, const
SCAInt64 lst1, const
SCAInt64 lfin1, const DynReal64& eplas,const DynReal64& effsts, const DynReal64& pres,const DynReal64& sie,const
DynReal64& rho,
DynReal64& damt, DynReal64& softe, DynReal64& volpls ,const DynReal64& depi, DynReal64& sxx , DynReal64& syy
,DynReal64&  szz ,
DynReal64& sxy , DynReal64& syz , DynReal64& sxz ,const DynReal64& sxxo, const DynReal64& syyo, const DynReal64&
szzo, const
DynReal64& sxyo, const DynReal64& syzo, const DynReal64& sxzo, const DynReal64& shear)
{
return SCASuccess;
}

SCAResult Materials::usrMatshr(const SCAInt64 id, const SCAInt64 lenvec, const DynReal64& exx, const DynReal64& eyy,
const DynReal64& ezz, const DynReal64& exy, const DynReal64& eyz, const DynReal64& ezx, const DynReal64& pres, const
DynReal64& edis, const DynReal64& sie, const DynReal64& rho, const DynReal64& fburn, const DynReal64& zmass,
DynReal64& shear, SCAInt64 ibegin, SCAInt64 iend)
SCAResult EulMat::usrMatshr(const SCAString flname, const SCAInt64 lennam, const SCAInt64 lenvec, const DynReal64&
exx, const DynReal64& eyy, const DynReal64& ezz, const DynReal64& exy, const DynReal64& eyz, const DynReal64& ezx,
const DynReal64& pres, const DynReal64& edis, const DynReal64& sie, const DynReal64& rho, const DynReal64& fburn,
const DynReal64& zmass, DynReal64& shear, SCAInt64 lst1, SCAInt64 lfin1)
{
return SCASuccess;
}
```

FORTRAN

```
      subroutine EXT_COMP  (id,lenvec,ishrpr,
     +                      ymx,ymy,xnuy,sxy,syz,szx
     +                      xt, xc, yt, yc, sc,
     +                      capa,cm, ibegin,
     +                      iend, nadvar, isubly,
     +                      sig1,sig2,sig4,
     +                      sig5,sig6,sig51,sig62,
     +                      d1,d2,d3,d4,d5,d6,d7,d8,
     +                      dout1, dout2, dout4
     +                      eft, efc, esf, emt, emc, q1, q2,
     +                      einc,fcap,
     +                      fail, falnm1,efail,fail2,user,
     +                      efib, emtx, eshr, lmcmar, err_code,
)
c
      implicit none
      integer(8),intent(in) :: id,lenvec,ishrpr,ibegin,iend
      integer(8),intent(in) :: nadvar, isubly, lmcmar
      integer, intent(inout) :: err_code
      real(8), intent(in) :: ymx, ymy, xnuy, sxy, syz, szx
      real(8), intent(in) :: xt, xc, yt, yc, sc, capa
      real(8), intent(in), dimension(lmcmar):: cm
      real(8), intent(in), dimension(lenvec):: d1, d2, d3, d4, d5, d6
      real(8), intent(in), dimension(lenvec):: d7, d8, dout1, dout2
      real(8), intent(in), dimension(lenvec):: dout4, q1, q2
      real(8), intent(in), dimension(lenvec,4):: fcap
      real(8), intent(inout), dimension(lenvec):: sig1, sig2, sig4
      real(8), intent(inout), dimension(lenvec):: sig5, sig6, sig51
      real(8), intent(inout), dimension(lenvec):: sig62, eft, efc, esf
      real(8), intent(inout), dimension(lenvec):: efib, emtx, eshr
      real(8), intent(inout), dimension(lenvec):: emt, emc, einc, fail
      real(8), intent(inout), dimension(lenvec):: falnm1, efail, fail2
      real(8), intent(inout), dimension(lenvec,nadvar):: user

c
      return
      end subroutine
```

id	material id.
lenvec	length of arrays.
ishrpr	transverse shear correction index (TRANSHR in PSHELL1)
	0 : constant transverse shear correction
	2 : linear transverse shear correction
	3 : approximated transverse shear correction
ymx	Young's modulus in fiber dir.
ymy	Young's modulus in matrix dir.
xnuy	Poisson's ration nuyx.
sxy	inplane shear modulus
syz	transverse shear modulus
szx	transverse shear modulus

xt	fiber tensile strength
xc	fiber compressive strength
yt	matrix tensile strength
yc	matrix compressive strength
sc	shear strength
capa	shear correction factor
cm	material constants used by input
cm(1)	E1 in MAT8
cm(2)	E2 in MAT8
cm(3)	not used
cm(4)	NU12 in MAT8
cm(5)	not used
cm(6)	not used
cm(7)	G12 in MAT8
cm(8)	G1Z in MAT8
cm(9)	G2Z in MAT8
cm(10)	MID
cm(11)	property degraing usage
	0.0 : no degraing
	1.0 : degraing option is used
cm(12)	PFDST in MAT8A
cm(13)	TRSFAIL in MAT8A
	0.0 : SUBL
	1.0 : ELEM
cm(14)	PRDFT in MAT8A
cm(15)	PRDFC in MAT8A
cm(16)	PRDMT in MAT8A
cm(17)	PRDMC in MAT8A
cm(18)	PRDSH in MAT8A
cm(19)	F12 in MAT8A
cm(20)	not used
cm(21)	sound speed number: sound speed is sqrt(cm(21)/cm(50))
cm(22)	not used
cm(23)	not used

cm(24)	not used
cm(25)	not used
cm(26)	S in MAT8A
cm(27)	XT in MAT8A
cm(28)	YT in MAT8A
cm(29)	YC in MAT8A
cm(30)	ALPHA in MAT8A
cm(31)	XC in MAT8A
cm(32)	PFD in MAT8A
	1.0: STEPS
	2.0: TIME
	3.0: VELOC
cm(33)	VALUE in MAT8A
cm(34)	FBTEN in MAT8A
	1.0: CHANG
	2.0: HILL
	3.0: TSAI
	4.0: MODTSAI
	5.0: STRESS
	6.0: USER
	7.0: COMBINAT
	8.0: HASHIN
	9.0: USER1
cm(35)	FBCOM in MAT8A. value is identical as cm(34)
cm(36)	MXTEN in MAT8A. value is identical as cm(34)
cm(37)	MXCOM in MAT8A. value is identical as cm(34)
cm(38)	MXSHR in MAT8A. value is identical as cm(34)
cm(39)-cm(49)	not used
cm(50)	RHO in MAT8
cm(51)	not used
cm(52)	internal material type: 22.0
cm(53)-cm(57)	not used
cm(58)	output system
	0.0: element system

	1.0: fiber system
cm(59)	usage of FT. It is always set to 1.0
cm(60)	FT in MAT8A. It is always set to 9.0
cm(61)	not used
ibegin	begin element counter
iend	end element counter
nadvar	number of additional vars
	(see mat8a bulk data)
isubly	sublayer number
sig1	sigma xx in fiber system
sig2	sigma yy in fiber system
sig4	sigma xy in fiber system
sig5	sigma yz in fiber system
sig6	sigma zx in fiber system
sig51	additional sigma yz due to linear shear correction
sig62	additional sigma zx due to linear shear correction
d1	strain increment xx
d2	strain increment yy
d3	strain increment zz
d4	shear angle = 2.0 x strain increment xy
d5	strain increment yz
d6	strain increment zx
d7	additional strain increment yz due to linear transverse shear correction
d8	additional strain increment zx due to linear transverse shear correction
dout1	total xx-strain for output
dout2	total yy-strain for output
dout4	total xy-strain for output
eft	fiber tension fail switch
	0: fail
	1: not fail
eft	fiber compression fail switch
	0: fail
	1: not fail
esf	shear fail switch

	0: fail
	1: not fail
emt	matrix tenstion fail switch
	0: fail
	1: not fail
emc	matrix compression fail switch
	0: fail
	1: not fail
q1	fiber axis rel to element sys
q2	matrix axis
einc	internal energy increment
fcap	for transverse shear correction
fail	overall element fail switch
falnm1	transverse shear failure index only available when TRSFAIL in MAT8A is set to ELEM
	0: fail
	between 0 to 1: degrading transverse shear stress
	1: not fail
efail	complete element failure
	0.0: fail
	1.0: not fail
fail2	one-dimensional time step suppression
user	user history variables. the size is LENVEC,NADVAR
efib	property degradation fiber mode
emtx	property degradation matrix mode
eshr	property degradation shear mode
lmcmar	size of cm array - currently 61
err_code	the returned error code
	0: success
	otherwise: fail

```
subroutine EXT_MATYLD(id,lenvec,iusrzn,
     +                      pres,edis,
     +                      sie,rho,fburn,effpls,zmass,effsts,twoj2,
     +                      effsr,usrvr1,usrvr2,relv,
     +                      sxxo,syyo,szzo,sxyo,syzo,sxzo,
```

```
     +                       sxxt,syyt,szzt,sxyt,syzt,sxzt,
     +                       dexx,deyy,dezz,dexy,deyz,dezx,
     +                       tdet,yldsq,damage,softe,volpls,
     +                       ibegin,iend,err_code)
c
      implicit none
      integer(8),intent(in) :: id, lenvec, ibegin,iend
      integer, intent(inout) :: err_code
      integer(8), intent(in), dimension(lenvec):: iusrzn
      real(8), intent(in), dimension(lenvec):: pres,edis
      real(8), intent(in), dimension(lenvec):: sie,rho,fburn
      real(8), intent(in), dimension(lenvec):: zmass,effsts
      real(8), intent(in), dimension(lenvec):: twoj2,effsr,relv
      real(8), intent(in), dimension(lenvec):: sxxo,syyo,szzo
      real(8), intent(in), dimension(lenvec):: sxyo,syzo,sxzo
      real(8), intent(in), dimension(lenvec):: sxxt,syyt,szzt
      real(8), intent(in), dimension(lenvec):: sxyt,syzt,sxzt
      real(8), intent(in), dimension(lenvec):: dexx,deyy,dezz
      real(8), intent(in), dimension(lenvec):: dexy,deyz,dezx
      real(8), intent(inout), dimension(lenvec):: usrvr1,usrvr2,effpls
      real(8), intent(inout), dimension(lenvec):: tdet,yldsq,damage
      real(8), intent(inout), dimension(lenvec):: softe,volpls

c
      return
      end subroutine
```

id	id on YLDEX entry. Is given by the value of "yld" on the YLDEX entry.
lenvec	length of arrays.
iusrzn	element number.
pres	old element pressure.
edis	old element distortional energy.
sie	old element specific internal energy.
rho	element density.
fburn	(euler) burn fraction of element.
effpls	element effective plastic strain.
zmass	mass of element.
effsts	element old effective stress.
twoj2	tentative second invariant element.
effsr	effective strain rate element.
usrvr1,usrvr2	user variables element.
relv	relative element volume.
sxxo..sxzo	old deviatoric element stresses.
sxxt..sxzt	trial deviatoric element stresses.
dexx..dezx	strain rate components.

tdet	detonation time element.
yldsq	yield stress.
damage	array that allows the user to store an additional variable. Contents are transported and clumped. This array can also be accessed within exyld and exfail2.
softe	same use as damage.
volpls	same use as damage.
ibegin	start of the element loop.
iend	end of the element loop.
err_code	is the returned error code; 0 = success, otherwise fail.

```
      subroutine EXT_MATEOS
     +           (id,lenvec,ibegin,iend,
     +            rho,dv,devis,xmass,fburn,pold,sieold,
     +            pnew,sienew,clnew,grungm,damage,
     +            softe,volpls,err_code)
c
       implicit none
       integer(8), intent(in) :: id, lenvec, ibegin, iend
       integer(8), intent(inout) :: err_code
       real(8), dimension(lenvec), intent(in):: rho,dv,devis,xmass
       real(8), dimension(lenvec), intent(in):: pold,sieold,fburn
       real(8), dimension(lenvec), intent(inout):: pnew,sienew
       real(8), dimension(lenvec), intent(inout):: clnew,grungm
       real(8), dimension(lenvec), intent(inout):: damage,softe,volpls
c
       return
       end subroutine
```

id	id on EOSEX entry. Is given by the value of "eid" on the EOSEX entry.
lenvec	length of arrays.
ibegin	start of the element loop.
iend	end of the element loop.
rho	element density.
dv	volume change of element.
devis	viscous work term.
xmass	mass of element.
fburn	burn fraction of element (not type 8).
pold	old pressure of element.
sieold	old specific internal energy of element.
pnew	new pressure of element.
sienew	new specific internal energy of element.

clnew	new sound speed of element.
grungm	Gruneisen gamma of element (only type 11).
	p = a*f(rho/rhoref) + b*f(rho*sie)
	where b = Gruneisen gamma.
damage	array that allows the user to store an additional variable. Contents are transported and clumped. This array can also be accessed within EXT_MATYLD and EXT_MATFAIL2.
softe	same use as damage.
vlpl	same use as damage.
err_code	is the is returned error code; 0 = success, otherwise fail

```
      subroutine EXT_MATFAIL
     + (id, lenvec,
     +  ibegin,iend,eplas,effsts,pres,sie,rho,ffail,iusrzn,err_code)
c
      implicit none
      integer(8), intent(in):: id, lenvec,ibegin,iend
      integer(8), intent(inout)::  err_code
      real(8), intent(in), dimension(lenvec):: eplas,effsts
      real(8), intent(in),  dimension(lenvec):: pres,sie,rho
      real(8), intent(inout), dimension(lenvec):: ffail
      integer(8), dimension(lenvec):: iusrzn
c
      end subroutine
```

id	id on EXFAIL entry. Is given by the value of "fid" on the FAILEX entry.
lenvec	length of arrays.
ibegin	start of the element loop.
iend	end of the element loop.
eplas	element effective plastic strain.
effsts	element effective stress.
pres	element pressure.
sie	element specific internal energy.
rho	element density.
iusrzn	user numbers of elements.
err_code	is the is returned error code; 0 = success, otherwise fail

```
      subroutine EXT_MATFAIL1(id, lenvec,
     + ibegin,iend,iusrzn,txx,tyy,tzz,txy,tyz,txz,depxx,
     + depyy,depzz,depxy,depyz,depxz,epxx,epyy,epzz,epxy,epyz,epxz,
     + exx,eyy,ezz,exy,eyz,exz,usrvr1,usrvr2,tstepzn,ffail,err_code)
c
      implicit none
```

```
      integer(8), intent(in):: id, lenvec,ibegin,iend
      integer(8), intent(in),dimension(lenvec):: iusrzn
      integer(8), intent(inout)::  err_code
      real(8), intent(in),dimension(lenvec):: txx,tyy,tzz,txy,tyz,txz,depxx
      real(8), intent(in),dimension(lenvec):: depyy,depzz,depxy,depyz,depxz
      real(8), intent(in),dimension(lenvec):: epxx,epyy,epzz,epxy,epyz,epxz
      real(8), intent(in),dimension(lenvec):: exx,eyy,ezz,exy,eyz,exz
      real(8), intent(in),dimension(lenvec):: usrvr1,usrvr2
      real(8), intent(inout),dimension(lenvec):: tstepzn,ffail
c
      end subroutine
```

id	id on FAILEX entry. Is given by the value of "fid" on the FAILEX entry.
lenvec	length of arrays.
ibegin	begin element counter.
iend	end element counter.
iusrzn	user numbers of elements.
txx...tzz	element normal stress components.
txy...tyz	element shear stress components.
depsxx...depszz	element normal strain increments.
depsxy...depsyz	element shear strain increments.
epsxx...epszz	element normal (last cycle) strain components.
epsxy...epsyz	element shear (last cycle) strain components.
exx...gzx	element elasticity matrix components.
sxx..sxz	deviatoric stress tensor of the current cycle after radial scale back.
usrvr1, usrvr2	element user variables.
tstepzn	element time step.
ffail	element failure flag:
	0 : element has failed
	1 : element has not failed
err_code	is the returned error code; 0 = success, otherwise fail.

```
subroutine EXT_MATFAIL2(id, lenvec,
     + ibegin,iend,eplas,effsts,pres,sie,rho,damage,
     + softe,volpls,depi,
     + sxx, syy, szz, sxy, syz, sxz,
     + sxxo, syyo, szzo, sxyo, syzo, sxzo,
     + shear,err_code)
c
      implicit none
      integer(8), intent(in):: id, lenvec,ibegin,iend
      integer(8), intent(inout)::  err_code
      real(8), intent(in),dimension(lenvec):: eplas,effsts,pres
      real(8), intent(in),dimension(lenvec):: depi,sie,rho,shear
      real(8), intent(in),dimension(lenvec):: sxxo, syyo, szzo
```

```
      real(8), intent(in),dimension(lenvec):: sxyo, syzo, sxzo
      real(8), intent(inout),dimension(lenvec)::sxx,syy,szz,sxy,syz,sxz
      real(8), intent(inout),dimension(lenvec)::damage,softe,volpls

c
      end subroutine
```

id	id on FAILEX entry. Is given by the value of "fid" on the FAILEX entry.
lenvec	length of arrays.
ibegin	begin element counter.
iend	end element counter.
eplas	element effective plastic strain.
effsts	element effective stress.
pres	old element pressure.
sie	old element specific internal energy.
rho	element density.
damage	array that allows the user to store an additional variable. Contents are transported and clumped. This array can also be accessed within EXYLD and EXFAIL2.
softe	same use as damage.
volpls	same use as damage.
depi	plastic strain increment of the current cycle.
sxx..sxz	deviatoric stress tensor of the current cycle after radial scale back.
sxx0..sxzo	deviatoric stress tensor from the previous cycle.
shear	shear modulus.
err_code	is the returned error code; 0 = success, otherwise fail.

```
     subroutine EXT_MATSHR( id,lenvec,
     +                      exx,eyy,ezz,exy,eyz,ezx,
     +                      pres,edis,sie,rho,fburn,zmass,shear,
     +                      ibegin,iend,err_code)
c
      implicit none
      integer(8),intent(in) :: id, lenvec, ibegin, iend
      integer, intent(inout) :: err_code
      real(8), intent(in),dimension(lenvec):: exx,eyy,ezz,exy,eyz,ezx
      real(8), intent(in),dimension(lenvec):: pres,edis,sie,rho
      real(8), intent(in),dimension(lenvec):: fburn,zmass
      real(8), intent(inout),dimension(lenvec):: shear
c
      return
      end subroutine
```

id	id on exshr entry given by the value of "sid" on the SHREX entry.
lenvec	length of arrays.
exx..ezx	element strain components.
pres	element pressure.
edis	element distortional energy.
sie	element specific internal energy.
rho	element density.
fburn	element burn fraction.
zmass	mass of element.
shear	shear modulus.
ibegin	begin of the element loop.
iend	end of the element loop.
err_code	is the is returned error code, 0 = success, otherwise fail.

Properties User-defined Property Definition in a Dytran Simulation

Description

The Property user subroutine is provided to allow the user to define a spring or damper model in a Dytran simulation. Properties permits the user definition by user subroutines:

1. EXT_PROPELAS: User defined Elastic Spring model using PELASEX.
2. EXT_PROPSPR: User defined Linear-Elastic Spring model using PSPREX.
3. EXT_PROPVISC: User defined Linear-Damper model using PVISCEX.

Format

IDL

```
module SCA {
module MDSolver {
module Obj {
module Uds {
module Dytran {
module Properties {
typedef SCAInt64 DynInt64[];
typedef SCAReal64 DynReal64[];

interface SCAIMDSolver700Properties : SCAIService
{
SCAResult usrPropelas(in SCAInt64 n,
in SCAInt64 m,
in DynInt64 ix,
in DynInt64 ic,
in DynReal64 prop,
in DynReal64 hisv,
in SCAReal64 forceo,
in DynReal64 c,
in DynReal64 di,
in DynReal64 v,
in SCAReal64 urel,
in SCAReal64 durel,
in SCAReal64 vrel,
in DynReal64 xmass,
out SCAReal64 force,
out SCAReal64 stiff
);
SCAResult usrPropspr (in SCAInt64  n,
in SCAInt64  m,
in DynInt64  ix,
in DynInt64  ic,
in DynReal64 prop,
in DynReal64 hisv,
in SCAReal64 forceo,
in DynReal64 c,
in DynReal64 di,
in DynReal64 v,
in SCAReal64 urel,
```

```
in SCAReal64 durel,
in SCAReal64 vrel,
in DynReal64 xmass,
out SCAReal64 force,
out SCAReal64 stiff
);
SCAResult usrPropvisc(in SCAInt64 n,
in SCAInt64 m,
in DynInt64 ix,
in DynInt64 ic,
in DynReal64 prop,
in DynReal64 hisv,
in SCAReal64 forceo,
in DynReal64 c,
in DynReal64 di,
in DynReal64 v,
in SCAReal64 urel,
in SCAReal64 durel,
in SCAReal64 vrel,
in DynReal64 xmass,
out SCAReal64 force
);
};
}; //Properties
}; //Dytran
}; //Uds
}; //Obj
}; //MDSolver
}; //SCA
```

C++

```
SCAResult Properties::usrPropelas(const SCAInt64 n, const SCAInt64 m, const DynInt64&
ix, const DynInt64& ic, const DynReal64& prop, const DynReal64& hisv, const SCAReal64
forceo, const DynReal64& c, const DynReal64& di, const DynReal64& v, const SCAReal64
urel, const SCAReal64 durel, const SCAReal64 vrel, const DynReal64& xmass, SCAReal64&
force, SCAReal64& stiff)
{
return SCASuccess;
}
SCAResult Properties::usrPropspr(const SCAInt64 n, const SCAInt64 m, const DynInt64&
ix, const DynInt64& ic, const DynReal64& prop, const DynReal64& hisv, const SCAReal64
forceo, const DynReal64& c, const DynReal64& di, const DynReal64& v, const SCAReal64
urel, const SCAReal64 durel, const SCAReal64 vrel, const DynReal64& xmass, SCAReal64&
force, SCAReal64& stiff)
{
return SCASuccess;
}
SCAResult Properties::usrPropvisc(const SCAInt64 n, const SCAInt64 m, const DynInt64&
ix, const DynInt64& ic, const DynReal64& prop, const DynReal64& hisv, const SCAReal64
forceo, const DynReal64& c, const DynReal64& di, const DynReal64& v, const SCAReal64
urel, const SCAReal64 durel, const SCAReal64 vrel, const DynReal64& xmass, SCAReal64&
force)
{
return SCASuccess;
}
```

FORTRAN

```
      subroutine EXT_PROPELAS(n,m,ix,ic,prop,hisv,
     +                        forceo,c,di,v,urel,durel,
     +                        vrel,xmass,force,stiff,
     +                        err_code)
c
```

```
      implicit none
c
      integer(8),intent(in)                 :: n,m
      integer(8),intent(in),dimension(2)    :: ix, ic
      real(8)   ,intent(in)                 :: forceo,urel,durel,vrel
      real(8)   ,intent(in),dimension(6)    :: prop
      real(8)   ,intent(in),dimension(6)    :: hisv
      real(8)   ,intent(in),dimension(3,2)  :: c
      real(8)   ,intent(in),dimension(6,2)  :: di,v
      real(8)   ,intent(in),dimension(2)    :: xmass
      real(8)   ,intent(out)                :: force,stiff
      integer   ,intent(inout)              :: err_code
c
      return
      end subroutine
```

input:	
N	Element id
M	Property id
IX(2)	Connectivity: IX(1) =grid point at end 1
IC(2)	Component:
IC(1)	component at end 1 (between 1 and 6)
IC(2)	component at end 2 (between 1 and 6)
PROP(6)	Properties as input on the PSPREX entry
HISV(6)	History variables for the element. This array can be used by the user to store variables from one time step to the next.
FORCEO	Force in the element at the previous time step
C(3,2)	Deformed coordinates in the basic coordinate system:
C(1:3,1)	x, y, z, coordinates at end 1
C(1:3,2)	x, y, z, coordinates at end 2
DI (6,2)	Incremental displacements in the basic coordinate system
DI(1:3,1)	x, y, z, translational displacements of end 1
DI(4:6,1)	x, y, z, rotational displacements of end 1
DI(1:3,2)	x, y, z, translational displacements of end 2
DI(4:6,2)	x, y, z, rotational displacements 2
	These are incremental displacements; i.e., the displacements for this time step only.
V(6,2)	Velocities in the basic coordinate system
V(1:3,1)	x, y, z, translational velocities of end 1
V(4:6,1)	x, y, z, rotational velocities of end 1

input:	
V(1:3,2)	x, y, z, translational velocities of end 2
V(4:6,2)	x, y, z, rotational velocities of end 2
A(6,2)	Accelerations in the basic coordinate system
A(1:3,1)	x, y, z translational accelerations of end 1
A(4:6,1)	x, y, z rotational accelerations of end 1
A(1:3,2)	x, y, z, translational accelerations of end 2
A(4:6,2)	x, y, z, rotational accelerations of end 2
UREL	Relative displacement of the element; i.e., the displacement of end 2 in the spring direction minus the displacement of end 1
DUREL	Relative incremental displacement of the element VREL
VREL	Relative velocity of the end points of the element in the direction of the element
XMASS(2)	Mass of the grid points at ends 1 and 2

output:	
force	Force in the element
stiff	Current stiffness of the element
err_code	returned error code; 0=success, otherwise fail

```
      subroutine EXT_PROPSPR(n,m,ix,ic,prop,hisv,
     +                       forceo,c,di,v,urel,durel,
     +                       vrel,xmass,force,stiff,
     +                       err_code)
c
      implicit none
c
      integer(8),intent(in)                :: n,m
      integer(8),intent(in),dimension(2)   :: ix, ic
      real(8)   ,intent(in)                :: forceo,urel,durel,vrel
      real(8)   ,intent(in),dimension(6)   :: prop
      real(8)   ,intent(in),dimension(6)   :: hisv
      real(8)   ,intent(in),dimension(3,2):: c
      real(8)   ,intent(in),dimension(6,2):: di,v
      real(8)   ,intent(in),dimension(2)   :: xmass
      real(8)   ,intent(out)               :: force,stiff
      integer   ,intent(inout)             :: err_code
c
      return
      end subroutine
```

input:	
N	Element id
M	Property id
IX(2)	Connectivity: IX(1) = grid point at end 1
IC(2)	Component:
IC(1)	component at end 1 (between 1 and 6)
IC(2)	component at end 2 (between 1 and 6)
PROP(6)	Properties as input on the PSPREX entry
HISV(6)	History variables for the element. This array can be used by the user to store variables from one time step to the next.
FORCEO	Force in the element at the previous time step
C(3,2)	Deformed coordinates in the basic coordinate system:
C(1:3,1)	x, y, z, coordinates at end 1
C(1:3,2)	x, y, z, coordinates at end 2
DI (6,2)	Incremental displacements in the basic coordinate system
DI(1:3,1)	x, y, z, translational displacements of end 1
DI(4:6,1)	x, y, z, rotational displacements of end 1
DI(1:3,2)	x, y, z, translational displacements of end 2
DI(4:6,2)	x, y, z, rotational displacements 2
	These are incremental displacements; i.e., the displacements for this time step only.
V(6,2)	Velocities in the basic coordinate system
V(1:3,1)	x, y, z, translational velocities of end 1
V(4:6,1)	x, y, z, rotational velocities of end 1
V(1:3,2)	x, y, z, translational velocities of end 2
V(4:6,2)	x, y, z, rotational velocities of end 2
A(6,2)	Accelerations in the basic coordinate system
A(1:3,1)	x, y, z translational accelerations of end 1
A(4:6,1)	x, y, z rotational accelerations of end 1
A(1:3,2)	x, y, z, translational accelerations of end 2
A(4:6,2)	x, y, z, rotational accelerations of end 2
UREL	Relative displacement of the element; i.e., the displacement of end 2 in the spring direction minus the displacement of end 1
DUREL	Relative incremental displacement of the element VREL

input:	
VREL	Relative velocity of the end points of the element in the direction of the element
XMASS(2)	Mass of the grid points at ends 1 and 2

output:	
force	Force in the element
stiff	Current stiffness of the element
err_code	returned error code; 0 = success, otherwise fail

```
      subroutine EXT_PROPVISC(n,m,ix,ic,prop,hisv,
     +                        forceo,c,di,v,urel,durel,
     +                        vrel,xmass,force,
     +                        err_code)
c
      implicit none
c
      integer(8),intent(in)                :: n,m
      integer(8),intent(in),dimension(2)   :: ix, ic
      real(8)   ,intent(in)                :: forceo,urel,durel,vrel
      real(8)   ,intent(in),dimension(6)   :: prop
      real(8)   ,intent(in),dimension(6)   :: hisv
      real(8)   ,intent(in),dimension(3,2):: c
      real(8)   ,intent(in),dimension(6,2):: di,v
      real(8)   ,intent(in),dimension(2)   :: xmass
      real(8)   ,intent(out)               :: force
      integer   ,intent(inout)             :: err_code
c
      return
      end subroutine
```

input:	
N	Element id
M	Property id
IX(2)	Connectivity: IX(1) =grid point at end 1
IC(2)	Component:
IC(1)	component at end 1 (between 1 and 6)
IC(2)	component at end 2 (between 1 and 6)
PROP(6)	Properties as input on the PSPREX entry

input:	
HISV(6)	History variables for the element. This array can be used by the user to store variables from one time step to the next.
FORCEO	Force in the element at the previous time step
C(3,2)	Deformed coordinates in the basic coordinate system:
C(1:3,1)	x, y, z, coordinates at end 1
C(1:3,2)	x, y, z, coordinates at end 2
DI (6,2)	Incremental displacements in the basic coordinate system
DI(1:3,1)	x, y, z, translational displacements of end 1
DI(4:6,1)	x, y, z, rotational displacements of end 1
DI(1:3,2)	x, y, z, translational displacements of end 2
DI(4:6,2)	x, y, z, rotational displacements 2
	These are incremental displacements; i.e., the displacements for this time step only.
V(6,2)	Velocities in the basic coordinate system
V(1:3,1)	x, y, z, translational velocities of end 1
V(4:6,1)	x, y, z, rotational velocities of end 1
V(1:3,2)	x, y, z, translational velocities of end 2
V(4:6,2)	x, y, z, rotational velocities of end 2
A(6,2)	Accelerations in the basic coordinate system
A(1:3,1)	x, y, z translational accelerations of end 1
A(4:6,1)	x, y, z rotational accelerations of end 1
A(1:3,2)	x, y, z, translational accelerations of end 2
A(4:6,2)	x, y, z, rotational accelerations of end 2
UREL	Relative displacement of the element; i.e., the displacement of end 2 in the spring direction minus the displacement of end 1
DUREL	Relative incremental displacement of the element VREL
VREL	Relative velocity of the end points of the element in the direction of the element
XMASS(2)	Mass of the grid points at ends 1 and 2

output:	
force	Force in the element
err_code	returned error code; 0 = success, otherwise fail

Loads

User-defined Loads in a Dytran Simulation

Description

The `Loads` user subroutine is provided to allow the user to define tables and define a velocity of nodes on a structure in a Dytran simulation.

1. EXT_VEL: User defined enforced motion at grid points using FORCEEX.
2. EXT_FUNC: User defined an arbitrary function using TABLEEX.

Format

IDL

```
module SCA {
module SCA {
module MDSolver {
module Obj {
module Uds {
module Dytran {
module Loads {

typedef SCAInt64 DynInt64[];
typedef SCAReal64 DynReal64[];

interface SCAIMDSolver700Loads : SCAIService
{

    SCAResult usrExtvel(   in SCAInt64 ilid,
                           in SCAInt64 ngp,
                           inout SCAReal64 xpos,
                           inout SCAReal64 ypos,
                           inout SCAReal64 zpos,
                           inout SCAReal64 xvel,
                           inout SCAReal64 yvel,
                           inout SCAReal64 zvel,
                           inout SCAReal64 xavel,
                           inout SCAReal64 yavel,
                           inout SCAReal64 zavel,
                           inout SCAReal64 pmass);


     SCAResult usrFunc(    in SCAInt64 ilid,
                           in DynReal64 xval,
                           inout DynReal64 yval,
                           in SCAInt64 numval);

};

}; //Loads
```

```
}; //Dytran
}; //Uds
}; //Obj
}; //MDSolver
}; //SCA
```

C++

```
SCAResult Loads::usrExtvel(const SCAInt64 ilid, const SCAInt64 ngp, SCAReal64& xpos, SCAReal64& ypos, SCAReal64&
zpos, SCAReal64& xvel, SCAReal64& yvel, SCAReal64& zvel, SCAReal64& xavel, SCAReal64& yavel, SCAReal64& zavel,
SCAReal64& pmass
)
{
return SCASuccess;
}
SCAResult Loads::usrFunc(const SCAInt64 id, const DynReal64& xval, DynReal64& yval, const SCAInt64 numval
)
{
return SCASuccess;
}
```

FORTRAN

```
 subroutine EXT VEL(ilid,ngp,xpos,ypos,zpos,
     +              xvel,yvel,zvel,xavel,yavel,zavel,pmass,
     +              err_code)

c
      implicit none
      integer(8),intent(in) :: ilid, ngp
      integer, intent(inout) :: err_code

c
      real(8), intent(inout) :: xpos, ypos, zpos
      real(8), intent(inout) :: xvel, yvel, zvel
      real(8), intent(inout) :: xavel, yavel, zavel, pmass

c
      end subroutine
```

ilid	lid of the forceex entry.
ngp	user node id.
xpos	x coordinate of node.
ypos	y coordinate of node.
zpos	z coordinate of node.
xvel	x nodal velocity.
yvel	y nodal velocity.
zvel	z nodal velocity.
xavel	x nodal angular velocity.
yavel	y nodal angular velocity.
zavel	z nodal angular velocity.
pmass	nodal mass.
err_code	is the returned error code; 0 = success, otherwise fail.

```
      subroutine EXT_FUNC(id,xval,yval,numval,err_code)
c
      implicit none
      integer(8), intent(in):: id, numval
      integer(8), intent(inout)::  err_code
      real(8), intent(in)   ,dimension(numval):: xval
      real(8), intent(inout),dimension(numval):: yval
c
      return
      end subroutine
```

`id`	id of the function defined on input.
`xval(nmstr)`	x-value that the function requires (time).
`nmstr`	number of values.
`yval(nmstr)`	value to be returned by the function. Note that the y-value is multiplied by the scale factor defined on the load entry.
`err_code`	returned error code, 0 = success, otherwise fail.

Connections User-defined Connections in a Dytran Simulation

Description

The Connections user subroutine is provided to allow the user to define failure criteria for BJOIN connections in a Dytran simulation.

1. EXT_BRK: User defined failure criteria for BJOIN

Format

IDL

```
namespace SCA {
namespace MDSolver {
namespace Obj {
namespace Uds {
namespace Dytran {
namespace Connections {

typedef SCAInt64 DynInt64[];
typedef SCAReal64 DynReal64[];
interface SCAIMDSolver700Materials : SCAIService {
SCAResult Connections::usrExbrk(const SCAReal64 time,
const SCAInt64  icycle,
const SCAInt64  nmsets,
const DynInt64& ilist,
const DynInt64& isetm,
const DynInt64& ibjoin,
      DynInt64& ifail,
const DynReal64& fx1,
const DynReal64& fy1,
const DynReal64& fz1,
const DynReal64& fx2,
const DynReal64& fy2,
const DynReal64& fz2,
const DynReal64& xm1,
const DynReal64& ym1,
const DynReal64& zm1,
const DynReal64& xm2,
const DynReal64& ym2,
const DynReal64& zm2,
const DynReal64& fail1,
const DynReal64& fail2,
const DynReal64& fail3,
const DynReal64& fail4,
const DynReal64& fail5,
const DynReal64& fail6,
const DynInt64& iconn1,
const DynInt64& iconn2,
const DynReal64& effpls);
};
}; //Connections
```

```
}; //Dytran
}; //Uds
}; //Obj
}; //MDSolver
}; //SCA
```

C++

```
SCAResult Connections::usrExbrk(const SCAReal64 time, const SCAInt64 icycle, const
SCAInt64 nmsets, const DynInt64&     ilist, const DynInt64& isetm, const DynInt64&
ibjoin, DynInt64& ifail, const DynReal64& fx1, const DynReal64& fy1,     const
DynReal64& fz1, const DynReal64& fx2, const DynReal64& fy2, const DynReal64& fz2, const
DynReal64& xm1, cons    t DynReal64& ym1, const DynReal64& zm1, const DynReal64& xm2,
const DynReal64& ym2, const DynReal64& zm2, const Dyn    Real64& fail1, const
DynReal64& fail2, const DynReal64& fail3, const DynReal64& fail4, const DynReal64&
fail5, cons    t DynReal64& fail6, const DynInt64& iconn1, const DynInt64& iconn2,
const DynReal64& effpls)
```

FORTRAN

```
      SUBROUTINE EXT_BRK
     +          ( TIME,ICYCLE,NMSETS,ILIST,ISETM,IBJOIN,IFAIL,
     +            FX1,FY1,FZ1,FX2,FY2,FZ2,
     +            XM1,YM1,ZM1,XM2,YM2,ZM2,
     +            FAIL1,FAIL2,FAIL3,FAIL4,FAIL5,FAIL6,
     +            ICONN1,ICONN2 ,EFFPLS, ERR_CODE )

      IMPLICIT NONE
      REAL(8), INTENT(IN):: TIME
      INTEGER(8), INTENT(IN):: ICYCLE,NMSETS
      INTEGER(8), INTENT(IN), DIMENSION(NMSETS):: ILIST,ISETM,IBJOIN
      INTEGER(8), INTENT(INOUT), DIMENSION(NMSETS):: IFAIL
      INTEGER(8), INTENT(IN), DIMENSION(8,NMSETS):: ICONN1,ICONN2
      REAL(8), INTENT(IN), DIMENSION(NMSETS):: FX1,FY1,FZ1,FX2,FY2,FZ2
      REAL(8), INTENT(IN), DIMENSION(NMSETS):: XM1,YM1,ZM1,XM2,YM2,ZM2
      REAL(8), INTENT(IN), DIMENSION(NMSETS):: FAIL1,FAIL2,FAIL3
      REAL(8), INTENT(IN), DIMENSION(NMSETS):: FAIL4,FAIL5,FAIL6
      REAL(8), INTENT(IN), DIMENSION(2*NMSETS):: EFFPLS
      INTEGER(8), INTENT(INOUT) :: ERR_CODE
*     declare local data
      INTEGER(8):: I, J, N, NP, NEL
      INTEGER(8):: ICLAST, IELEM, NELEM, NZONE1, NZONE2, NZONE_LOC
      INTEGER(8):: NPAIR, ISETN, NSETS, NENTRY, NMSTRS
      INTEGER(8), DIMENSION(NMAX1):: ISETS
      INTEGER(8), DIMENSION(NMAX2):: IGROUP
      INTEGER(8), DIMENSION(NMAX2,4):: IELCON, IPPNUM
      REAL(8):: EFPSUM
      LOGICAL:: LTRUE, LFALSE, LDOIT, LFIRST
      LOGICAL, DIMENSION(NMAX2):: LFAIL, LARRY

      return
      end subroutine
```

time	present time in computation
icycle	time cycle at present
nmsets	number of join sets in the user defined string
ilist	set numbers in this string
isetm	set number of the collection of bjoins to which the bjoin belongs. Bjoints are grouped into sets.

ibjoin	If < 0 : negative of the bjjoin id. > 0 : crod-id of a splotweld element
ifail	global failure flag for the set
fx1	x-component force of point1 of pair
mx1	x-component moment of point1 of pair
fail1	x-component force failure switch
fail2	y-component force failure switch
fail3	z-component force failure switch
fail4	x-component moment failure switch
fail5	y-component moment failure switch
fail6	z-component moment failure switch
iconn1,iconn2	contain connectivity data for the first and the second grid point of the BJOIN pairs. Data concerns the grid point user number, the number of connected elements, and the connected element user numbers. iconn1(1,n) = user number of first point of join pair n iconn1(2,n) = number of connected elements to point 1 of set n iconn1(3,n) = user number of connected element … iconn1(8,n) = user number of connected element
effpls	effective plastic strain of elements. element ids are given by ICONN1 and ICONN2.

8 Diagnostic Messages

Whenever Dytran encounters invalid or inconsistent data in the input file or a problem is encountered during the analysis, a diagnostic message is printed.

Diagnostic messages produced during the initialization and solution are written to the `<jobname>_ERROR_SUMMARY-MSG` file. These messages normally indicate incorrect or inconsistent data and problems encountered during the solution.

Each diagnostic message is produced a maximum of five times to prevent large quantities of output from being produced.

The diagnostic message is a set of short codes that indicate the severity of the message, its number, and the subroutine that generated it. One or more lines of text follows, indicating what the problem is.

The coded line has the basic form:

```
%x-<diagnostic number>-<subroutine name>
```

where `x` indicates the severity:

- `I` Information
- `W` Warning
- `E` Error
- `F` Fatal
- `C` Catastrophic

Information messages do not indicate a problem, and the analysis should continue successfully.

Warnings are not fatal, and the analysis will continue. However, warnings are an indication that something about analysis is not normal. You should review all warnings carefully and make sure you know what is causing the message.

```
%W-P3007905-P3XXTXX_CYCLE_ZERO
Authorization files will expire this month
```

Error messages indicate that there is almost certainly something wrong with your analysis. You should review your input and modify it appropriately. Errors in the solution cause termination at the end of the current time step. If you specified output at the end of the analysis, then the output is produced before the analysis terminates.

```
%E-CN000602-SYS_CHECK_RST_INPUT_FILE
Restart input file must have extension .RST
```

Fatal messages have the same effect as error messages but indicate a more serious problem.

```
%F-2039502-P2XXTXIX_PACKET_EDIT
Materials cannot be put on archive files, only on time history files.
```

Catastrophic errors are issued only when the program would otherwise crash. They may occur, for example, when the analysis would result in a division by zero.

```
%C-P2057301-C2_STORE_IZONER_FOR_CFACE
Face number -4 is illegal
```

The severity code letter is followed by the unique diagnostic number. There is a unique number for each diagnostic message and it can be used to reference the message. The subroutine name indicates which routine produced the diagnostic message. Note that the routine names in Dytran consist of up to 31 characters.

Internal program errors have the message (PROGRAM ERROR) at the end of the first line of the diagnostic message, e.g.,

```
%E-CN000502-SYS_CHECK_RST_INPUT_FILE (PROGRAM ERROR)
RESTART INPUT FILE HAS WRONG EXTENSION
```

You should never get program errors. If you do, check if there are other diagnostic messages indicating other problems. If not, please note the diagnostic number and contact your local MSC representative.

Redefinition of severity and number of prints of diagnostic messages can be performed by using PARM,ERRUSR. This parameter has to be used carefully.

A References

1. *MSC/DYNA User's Manual*, The MacNeal-Schwendler Corporation.
2. *MSC/PISCES-2DELK User's Manual*, The MacNeal-Schwendler Corporation.
3. E. L. Lee et al., "Adiabatic Expansions of High Explosive Detonation Products", UCRL-50422, May 1968, Lawrence Livermore National Laboratory, Livermore, California.
4. *MADYMO User's Manual 3-D Version 4.3*, The Dutch Institute of Applied Scientific Research - Road/Vehicles Research Institute, Department of Injury Prevention, October 1988.
5. Louise A. Obergefell, Thomas R. Gardner, Ints Kaleps, and John T. Fleck, *Articulated Total Body Model Enhancements*, Volume 1: "Modifications", (NTIS No. ADA198726).
6. Louise A. Obergefell, Thomas R. Gardner, Ints Kaleps, and John T. Fleck, *Articulated Total Body Model Enhancements*, Volume 2: "User's Guide", (NTIS No. ADA203566).
7. Louise A. Obergefell, Thomas R. Gardner, Ints Kaleps, and John T. Fleck, *Articulated Total Body Model Enhancements*, Volume 3: "Programmer's Guide", (NTIS No. ADA197940).
8. J. T. Fleck, F. E. Butler, and N. J. Deleys, "Validation of the Crash Victim Simulator", Calspan Report Nos. ZS-5881-V-1 through 4, DOT-HS-806-279 through 282, 1982, Volumes 1 through 4, (NTIS No. PC E99, PB86-212420).
9. P. L. Roe, *Approximate Riemann Solvers, parameter vectors, and difference schemes,* Journal of Computational Physics, 43, 357-372, 1981.
10. P. L. Roe and J. Pike, *Efficient construction and utilisation of approximate Riemann solutions,* Computing Methods in Applied Sciences and Engineering (VI), R. Glowinski and J. L. Lions (Editors), Elsevier Publishers B.V. (North Holland), INRIA 1984.
11. Bram van Leer, Chang-Hsien Tai, and Kenneth G. Powell, *Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations,* AIAA-89-1933-CP, 1989.
12. Ch. Hirsch, *Numerical Computation of Internal and External Flows, Fundamentals of Numerical Discretization, 1*, Wiley, Chichester, 1988.
13. Ch. Hirsch, *Numerical Computation of Internal and External Flows, Computational Methods for Inviscid and Viscous Flows, 2,* Wiley, Chichester, 1990
14. Louise A. Obergefell, Huaing Cheng, Annnette L. Rizer, *Articulated Total Body Model Version V*: "User's Manual", (NTIS No. ASC-98-0807), 1998.
15. Louise A. Obergefell, Huaing Cheng, Annnette L. Rizer, *Input Description for the Articulated Total Body Model ATB-V.1*, 1998.
16. Joseph A. Pellettiere, Huaing Cheng, Annnette L. Rizer, *The Development of GEBOD Version V*, 2000.
17. S.Tanimura, K.Mimura, and W.H. Zhu, *Practical Constitutive Models Covering Wide Ranges of Strain Rates, Strains and Temperatures,* Key Engineering Materials Vols. 177-180,189-200, 2000
18. W.A. van der Veen, "Simulation of a compartmented airbag deployment using an explicit, coupled Euler/Lagrange method with adaptive Euler Domains", Presented at: Nafems 2003, Orlanda. Available at the Dytran section of www.mscsoftware.com.