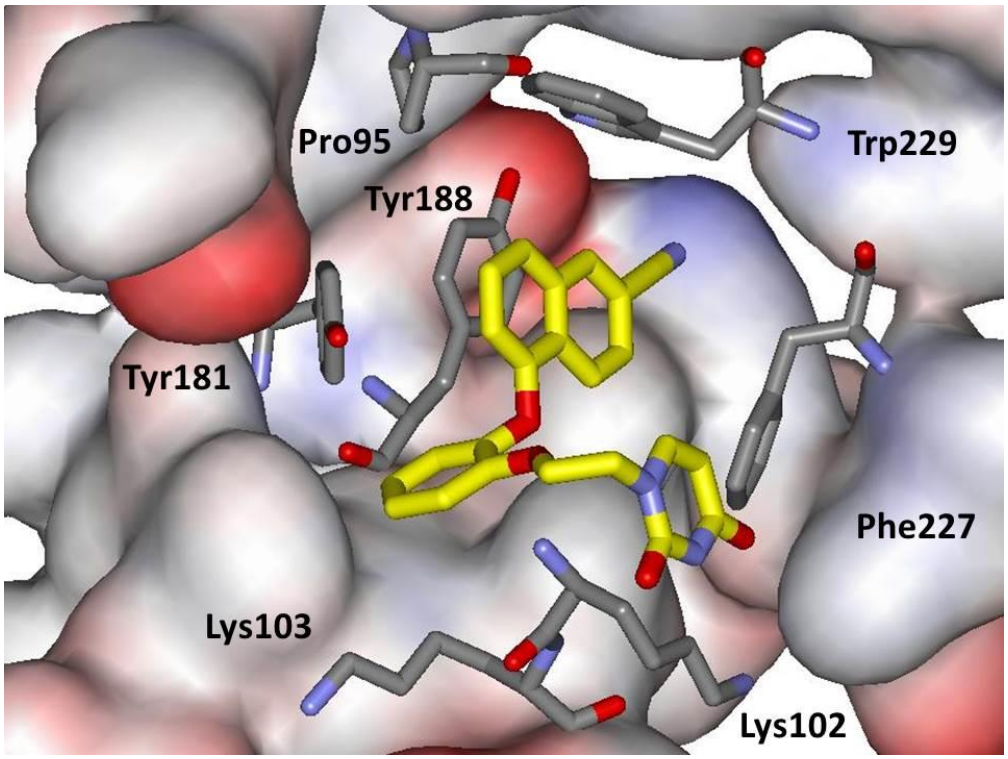
MCPRO, Version 2.3
Monte Carlo Simulations for Biomolecules

User's Manual



William L. Jorgensen and Julian Tirado-Rives
Department of Chemistry, Yale University
New Haven, Connecticut 06520-8107
email: william.jorgensen@yale.edu

June 2015

Contents

Monte Carlo Simulations for Proteins and Peptides

1 Introduction

The **MCPRO** program performs (a) **Monte Carlo (MC) statistical mechanics simulations** for solutions of zero to 25 solute molecules in a periodic solvent box, in a solvent cluster, or in a dielectric continuum including the gas phase and GB/SA model, and (b) **standard molecular mechanics energy minimizations** with the **OPLS force fields**. It was developed from the Biochemical and Organic Simulation System (BOSS) program and retains many of its features. MCPRO is designed for MC simulations of peptides and proteins; it makes extensive use of the concept of residues in the sampling algorithms and cut-off procedures. These features are not in BOSS, which is more for organic solutes.

1. For Monte Carlo simulations, the **NPT ensemble** is employed, though NVT simulations can also be performed by not allowing volume changes. For energy minimizations, several procedures are provided including **conjugate gradient** optimizations.

2. **Preferential sampling** may be used whereby the solutes and nearby solvent molecules are moved more frequently than distant solvent molecules. The biasing is based on $1/(r^2 + wkc)$ where *wkc* is a constant. See *J. Phys. Chem.* **87**, 5311–5312 (1983).

3. The solute coordinates can be provided by **Z-matrix or PDB input** and internal coordinates are used to sample solute geometries. Initial solvent coordinates can come from stored, equilibrated boxes of solvent. Boxes of various sizes are currently provided for ten solvents. Other boxes with up to 5000 solvent molecules can be automatically created**. Binary solvent mixtures** can also be treated.

4. **Free energy** changes are computed from statistical perturbation (**FEP**) theory. Applications can include computing free energies of solvation, reaction profiles, potentials of mean force (pmf's) including free energy profiles for rotation of a chosen dihedral angle, and relative or absolute free energies of binding for host/guest complexes. A linear response or linear interaction energy method may also be used to approximate solvation and binding free energies.

For examples of protein-ligand binding calculations with the MCPRO program, see:

- trypsin/benzamidines: FEP – *J. Phys.Chem.B* **101**, 9663 (1997)
- cyclophilin/cyclosporin: FEP – *Angew. Chem. Int. Ed. Engl.* **36**, 1466 (1997)
- thrombin/MD-805: linear response – *J. Med. Chem.* **40**, 1539 (1997)
- FKBP/ligands: FEP – *J. Med. Chem.* **41**, 3928 (1998)
- FKBP/ligands: linear response – *Bioorg. Med. Chem.* **7**, 851 (1999)
- COX-1/COX-2: FEP – *J. Am. Chem. Soc.* **122**, 9455 (2000)
- COX-2: linear response – *Bioorg. Med. Chem. Lett.* **12**, 267 (2002)
- HIV-RT/Sustiva/mutants: FEP – *J. Am. Chem. Soc.* **122**, 12898 (2000); *BMCL* **11**, 2799 (2001)
- HIV-RT/TMC125/mutants: FEP – *J. Am. Chem. Soc.* **125**, 6016 (2003)
- HIV-RT/Sustiva/K103N: linear response – *BMCL* **13**, 3337 (2003)
- HIV-RT/250 ligands: linear response – *J. Med. Chem.* **45**, 2970 (2002)
- thrombin/20 ligands: linear response – *J. Med. Chem.* **44**, 1043 (2001)
- factor Xa/60 ligands: LR and FEP – *J. Med. Chem.* **46**, 5691 (2003)
- CDK2, p38, Lck kinases/148 ligands: LR – *J. Med. Chem.* **47**, 2534 (2004)
- HIV-RT inhibitor optimization – *review: Acc. Chem. Res.* **42**, 724 (2009)
- recent lead optimization of NNRTIs – *J. Med. Chem.* **54**, 8582 (2011)

Overview: An extensive overview of MCPRO and its capabilities has been published in *J. Comput. Chem.* **26**, 1689-1700 (2005). The reprint is included in the *mcproman* directory as *BOSS-MCPRO.JCC05.pdf*

For examples of calculations with the BOSS or MCPRO programs, see:

- free energies of hydration - *J. Chem. Phys.* **83**, 3050 (1985);
  - *J. Am. Chem. Soc.* **121**, 4827 (1999)
- GB/SA model - *J. Phys. Chem. B* **108**, 16264 (2004)
- linear response calculations - *J. Phys. Org. Chem.***10**, 563 (1997);
  - *J. Am. Chem. Soc.* **122**, 2878 (2000)
- partition coefficients - *J. Phys. Chem.* **94**, 1683 (1990)
- relative acidities (pK'_as) - *J. Am. Chem. Soc.* **111**, 4190 (1989);
  - *J. Phys. Chem. A* **104**, 7625 (2000)
- pmfs for ion pairs - *J. Am. Chem. Soc.* **111**, 2507 (l989)
- for benzene dimer - *J. Am. Chem. Soc.* **112**, 4768 (1990)
- for amide dimers - *J. Am. Chem. Soc.* **111**, 3770 (1989)
- reaction profiles - *ACS Symp. Ser.* **721**, 74 (1999);
  - *J. Chem. Theory Comput.* **3**, 1412 (2007);
  - *J. Am. Chem. Soc.* **132**, 3097 (2010);
  - *J. Am. Chem. Soc.* **132**, 8766 (2010)
- clusters - *J. Chem. Phys.* **99**, 4233 (1993)
- conformational equilibria - *J. Phys. Chem.* **91**, 6083 (1987)
  - *J. Am. Chem. Soc.* **114**, 7535 (1992)
  - *J. Am. Chem. Soc.* **116**, 2199 (1994)
- host/guest binding - *J. Org. Chem.* **64**, 7439 (1999)
  - *J. Am. Chem. Soc.* **120**, 5104 (1998)
  - *Proc. Natl. Acad. Sci.* USA **90**, 1194 (1993)
- PDDG/PM3 - *J. Chem. Theory Comput.* **1**, 817 (2005);
  - *J. Phys. Chem. A* **110**, 13551 (2006)
- QM/MM calculations - *J. Phys. Chem. B* **102**, 1787 (1998); **106**, 8078 (2002)
  - *J. Am. Chem. Soc.* **126**, 9054 (2004)
  - *review: Acc. Chem. Res.* **43**, 142 (2010)
- ligand optimization - *review: Acc. Chem. Res.* **42**, 724 (2009)
- force fields - *review: Proc. Nat. Acad. Sci. USA* **102**, 6665 (2005);
  - *ions: J. Chem. Theory Comput.* **2**, 1499 (2006);
  - *polarization: J. Chem. Theory Comput.* **3**, 1987 (2007)
- FEP calculations - *review: J. Chem. Theory Comput.* **4**, 869 (2008)
- JAWS - *J. Phys. Chem. B* **113**, 13337 (2009);
  - *J. Am. Chem. Soc.* **131**, 15403 (2009)

5. The **OPLS** united-atom (*J. Am. Chem. Soc.* **110**, 1657 (1988)) and all-atom (*J. Am. Chem. Soc.* **118**, 11225 (1996)) potential functions are used with additional references given in the parameter file. Coverage currently includes numerous organic functional groups, all common peptide residues, and nucleoside bases. More parameters can easily be added to the parameter file. Many **unpublished OPLS parameters**, including torsional parameters for the united-atom force field designed to reproduce *ab initio* profiles (e.g., *J. Comp. Chem*. **16**, 984 (1995)) and all-atom parameters for numerous heterocycles, and other **pharmacologically important groups,** are also provided only to BOSS/MCPRO users.

6. Dihedral angles, bond angles and bond lengths can be sampled over during the simulations, *e.g*., a pmf could be determined for the separation of two flexible molecules, or binding of a flexible ligand to

a protein could be studied. Residues defined in the Z-matrix are used in sampling and cutoff procedures. Protein residues are usually built and numbered sequentially with the ***pepz* program** (Appendix 1).

7. **Harmonic restraints** can be included between pairs of solute atoms. This facilitates some free energy calculations for host/guest systems and can also be used for constrained optimizations. Atoms can also be restrained to points in Cartesian space.

9. Coordinate files may be **output in PDB format** for easy display with standard molecular graphics programs.

Can't Wait to Start: Use the scripts in the *scripts* directory and also try the test jobs.

2 Statistical Mechanics Simulations—Theory

Simulations of many particle systems commonly use Monte Carlo (MC) statistical mechanics or molecular dynamics (MD). Both methods typically employ a classical potential energy function for bond stretching, angle bending, torsions, and non-bonded interactions. For the OPLS-AA force field:

Bond stretching: $$E_{bond} = \sum_{bonds} K_r (r - r_{eq})^2$$

Angle bending: $$E_{angle} = \sum_{angles} K_\theta (\theta - \theta_{eq})^2$$

Torsion: $$E_{dih} = V_1(1+\cos\phi)/2 + V_2(1-\cos 2\phi)/2 + V_3(1+\cos 3\phi)/2 + V_4(1-\cos 4\phi)/2$$

Non-bonded: $$E_{ab} = \sum_{i}^{on\,a} \sum_{j}^{on\,b} [q_i q_j e^2 / r_{ij} + 4\varepsilon_{ij}(\sigma_{ij}^{12} / r_{ij}^{12} - \sigma_{ij}^{6} / r_{ij}^{6})] f_{ij}$$

f_{ij} = 0.5 if i, j are 1,4; otherwise, f_{ij} = 1.0

A MC simulation generates a new configuration by a set of random motions. The difference in energy between the new and the old configuration is used as a selection criterion by the Metropolis algorithm, which enforces a correct Boltzmann distribution of energies for the system at the desired temperature, and the procedure is then iteratively repeated. The resultant ensemble of configurations can be used to compute the structure and thermodynamic properties of the system at the specified conditions. Differences in free energies between two similar states of the system can also be calculated through statistical perturbation theory. In a MD simulation, on the other hand, the forces from the energy components in the three Cartesian directions acting on each atom are evaluated. The accelerations are calculated from these forces, and a trajectory is generated by the repeated numerical integration of the equations of motion over a period of time. The trajectory constitutes an ensemble of configurations that is formally equivalent to an ensemble generated via a MC simulation and is used in the same fashion.

This difference in methodology has significant implications in the use and application of the methods. The major algorithmic difference is that MC sampling can readily use internal coordinates, while MD works with independent atomic motions in Cartesian space. The consequences of this difference are many; in MD sampling is easier for large, flexible molecules since any coupling between different degrees of freedom is automatically handled by the algorithm. On the other hand, the integration time

step has to be very small in order to avoid numerical instability. There is a certain inefficiency in the sampling since motions that go uphill in energy are usually reversed by opposing restoring forces. Also, the algorithm makes the selective freezing of unimportant degrees of freedom cumbersome. In a MC calculation, the freezing of selected internal coordinates is easily accomplished by simply not allowing variations in the specified bonds, angles, or dihedrals; however, sampling for large flexible molecules can be problematic since degrees of freedom are treated, for the most part, as being completely uncoupled. An additional advantage of MC is that since the motion is by nature random it is possible to sample large regions of configurational space, by stepping over potential energy barriers. A paper comparing the efficiencies of MD and MC in simulations of liquid hexane has appeared: W. J. Jorgensen & J. Tirado-Rives, *J. Phys. Chem.* **100**, 14508 (1996). In this case, MC was found to be 2–4 times more efficient for conformational sampling.

Summaries on the theory and computations can be found in the following paper along with references to more detailed presentations: W. L. Jorgensen, *J. Phys. Chem.* **87**, 5304 (1983). See also: M. P. Allen and D. J. Tildesley, Computer Simulations of Liquids; Oxford U. Press: London, 1987.

3 Energy and Free Energy Evaluation

In MCPRO, the total potential energy of a system is given by eq 1

$$E = ESS + ESX + EXX + EPOL + EBND + EBC + EANG + EDIH + ENB + ECUT + EGBSA \quad (1)$$

where

ESS = the solvent–solvent energy,
ESX = the solvent–solute energy (and solvent cap restraint, if any),
EXX = the solute–solute (intersolute) energy,
EPOL = the solute-solute explicit polarization energy (optional),
EBND = the bond stretching energy for the solutes,
EBC = the energy for the interatomic harmonic constraints,
EANG = the angle bending energy for the solutes,
EDIH = the torsional energy for the solutes,
ENB = the >1,3 intramolecular non-bonded energy for the solutes, and
ECUT = the cutoff correction for the Lennard–Jones interactions neglected beyond the cutoff for non–aqueous solvents,
EGBSA = optional GB/SA free energy of solvation.

Free energy changes, if desired, are computed with statistical perturbation theory in a windowing format with double–wide or simple overlap sampling. ΔG (Gibbs) and ΔA (Helmholtz) are computed for NPT and NVT simulations, respectively. FEP calculations are not implemented for GB/SA solvation. Otherwise, the expression for ΔG is in eq 2,

$$\Delta G = G_j - G_i = -kT \, ln < exp((E_j - E_i)/kT) >_i \quad (2)$$

where the perturbation is from the reference system *i* to the perturbed system *j*. The mutation of A to B involves a scaling where a coupling parameter λ is used such that for $\lambda = 0$ the system is A and for $\lambda = 1$ it is B. Then for any geometrical variable or potential function parameters $Y(q, \sigma, \varepsilon)$,

$$Y_i = \lambda_i \, Y_B + (1 - \lambda_i) Y_A \quad (3)$$

eq 3 is used to scale the system from A to B. Thus, the simulation is run at a reference value λ corresponding to system *i* in eq 2. The program perturbs the system to two other values of λ corresponding to *j* and *k*, where typically $j < i < k$. This computation of two free energy increments in one simulation has been termed "double–wide sampling" (*J. Chem. Phys.* **83**, 3050 (1985)).

The number of free energy increments that need to be computed depends on the details of the overall perturbation, the solvent and T, P conditions. Many examples are available in the references given on page 3. For the conversion of ethane to methanol in water using the BOSS program, 5 increments were sufficient to give excellent precision (*J. Chem. Phys.* **83**, 3050 (1985)), while the annihilation of adenine or guanine in chloroform required 30–40 increments necessitating 15–20 simulations with double-wide sampling (Tetrahedron, **47**, 2491 (1991)). In a MCPRO simulation of a FK506 binding protein inhibitor in water, 13 simulations were used to remove a phenyl substituent.

The perturbations can also involve a reaction coordinate such as a dihedral angle or intermolecular distance. This is easily specified in the Z-matrix file by indicating the beginning and ending values of the variable corresponding to $\lambda = 0$ and 1. In this case there might be no change in the atom types (and potential function parameters). In MCPRO, the three λ values are referred to as RC0 for the reference solute and RC1 and RC2 for the perturbed solutes (page 15).

4 New Features

For the June 2015 release of MCPRO 2.3:

1. The OPLS-AA/M force field for peptides and proteins has been implemented.

For the June 2014 release of MCPRO 2.3:

1. Atom types OA and SA have also been added for aromatic O and S in furan, thiophene, thiazole, etc.
2. Improved torsional energetics for substituted heterocycles has been implemented through additions to oplsaa.par and the newatom types.
3. The program typeQLJ has been provided in the miscexec subdirectory. Enter typeQLJ and it will query for a pdb file; output is a file with the OPLS/CM1A non-bonded parameters.
4. The executables for clu, chop, and pepz are now in miscexec.
5. A 30-A water cap has been added in solbox. It has 3777 water molecules.
6. Dihedral angle distributions are no longer computed when the total number of dihedral angles is 50 or more. This shortens the up-file by 99% in typical protein cases. If such distributions are needed, they can be computed from the coordinates in the sav-file. Thus, old up-files can not be used with this version of MCPRO.

For the July 2012 release of MCPRO 2.3:

1. Treatment of halogen bonding is handled with the OPLS-AAx force field. A preprint is provided in the MCPROman directory and the new testjob HalogenBond gives examples. The *BOSS* program was also modified to obtain CM1A charges for systems with X sites.

2. The MM-GB/SA evaluation of free energies of binding has been implemented, including the use of Residue-dependent Dielectric. The MM-GBSA test job illustrates the use of this procedure on the binding of the UC-781 ad UC-10 HIV-RT inhibitors.

For the November 2010 release of MCPRO 2.2:

1. A water placement algorithm, JAWS, has been implemented. JAWS is used to locate hydration sites in a protein binding site and estimate their affinity for water. The procedure is helpful to optimize the water distribution around a ligand before executing MC/FEP calculations. See chapter 17 and the new test jobs in the folder *JAWS*. This required changes to the parameter and Z-matrix files. The utility programs *fixpf23* and *fixzm23* in the *miscexec* directory are included to convert previous versions of the parameter files and Z-matrices, respectively.
2. Conjugate gradient or steepest descent optimizations can be done while maintaining some residues "frozen" at their initial positions. See test job 34 for an example.
3. Simplified setups can be done through the interactive and query modes of *chop*. See Appendix 1 for details.
4. The utility program *clu* is now included in the distribution. It can be used to replace ligands. See appendix 2

For the January 2008 release of MCPRO 2.1:
1. Overlap sampling has been implemented as an alternative to double-wide sampling for FEP calculations. See the new test jobs *SOSgas*, *SOSaq*, and *SOS-Prot*. Also see the preprint OverlapSampling in the *MCPROman* directory.
2. The *FEP-ScanCl* test job has been added to illustrate a Cl → H FEP scan for a protein-ligand complex. The *FEPgas* and *FEPaq* test jobs have also been updated.

For the November 2006 release of MCPRO 2.05:
1. Solute-solute polarization is optionally treated for all energy minimizations except conjugate-gradient and all MC simulations including MC/FEP calculations. This is potentially particularly important for cation-π interactions. Inducible dipoles are placed on the polarizable atoms; they are determined from permanent charges only. See the testjob *polarize* for details and the script xPOPTF.

Polarization is invoked by declaring POLSCX = 1.0 on line 59 of the parameter (par) file. All provided par files have been modified. Previous users will likely need to modify old par files to be compatible - old par files often have some text on this line that is now in the field read for POLSCX.

2. Treatment of allenes has been added - see allen.z, allen1m.z, and allen2m.z in *%MCPROdir%/molecules/small*. Other additions are in *oplsaa.par* and *oplsaa.sb* including new alkali cation and halide ion parameters (K. Jensen and WLJ, *JCTC*, **2**, 1499 (2006)).

For the January 2006 release of MCPRO 2.00:
1. The Windows version has been overhauled. The new automatic installer will install MCPRO in any directory, which is designated by the environment variable *MCPROdir*, normally *c:\Program Files\MCPRO* or *c:\MCPRO*. The installer also provides an MCPRO shell window on the desktop. You can have as many copies of the MCPRO shell window open as you wish. The MCPRO shell has *%MCPROdir%* properly defined from the installation and the *%MCPROdir%\scripts* directory is in the execution path; thus, all MCPRO scripts are available in the MCPRO shell.

2. Specified dihedral angles can now be "flipped" randomly during MC simulations, i.e., occasionally, large changes in the dihedral angles are attempted as specified in the Z-matrix with a "flip" statement. This facilitates hopping between alternate conformers that may be separated by substantial barriers. See page 24 and the flip subdirectories in the *MCgas* test job.

The maximum number of solute atoms is 6000. The maximum number of solvent molecules is 5000. The maximum number of residues is 395.

New features in prior releases of MCPRO 2.00 are:

1. A GB/SA hydration model has been implemented for optimizations, single-point calculations, dihedral driving, conformational searching, and Monte Carlo simulations; FEP-GB/SA calculations have not been implemented yet. The GB/SA treatment is based on the model described in Qiu, D.; Shenkin, P. S.; Hollinger, F. P.; Still, W. C., *J. Phys. Chem. A*, **101**, 3005-3014 (1997). A key difference is OPLS-AA sigmas divided by two ($\sigma/2$) are used as the van der Waals radii, R_{vdW}, whereas Still et al. used OPLS-UA sigmas. The results for 65 molecules including the 35 molecules in Table 3 of their paper yield an average error of 0.67 kcal/mol. A reprint on our implementation is in the *MCPROman* directory. Optimized atomic ion parameters are in *oplsaa.par* entries 1100-1114. The ionic strength of the solution can be specified on line 29 of the par file; see the file *par.form* in the *mcpro* directory (Ref.: *Ann. Rev. Phys. Chem.* **51**, 129-152 (2000)).

Scripts that use GB/SA have been added including xSPGB, xOPTGB, and xMCGB. See the new MCGBSA and FoldProt test jobs.

2. The efficient procedure for backbone sampling developed by Jakob Ulmschneider has been implemented using a concerted rotation algorithm. To use, set NCONROT in the par file to the frequency of conrot attempts, e.g., every 4^{th} configuration. Copies of the publications on the theory and applications to the folding of three peptides are in the *MCPROman* directory. See the new FoldProt and FoldRNA test jobs.

3. The x scripts are now located in the *scripts* directory.

4. Computed atomic charges from the quantum methods (CM1, CM3, Mulliken) for equivalent, monovalent atoms are now symmetrized by BOSS. Thus, for example, in actetate ion the three methyl hydrogens will have the same charge as will the two oxygen. Formally and formerly, in the absence of the appropriate symmetry, charges for such atoms are computed to be different by any quantum method. This change simplifies the use of the quantum charges with force field calculations; it improves, for example, conformational searching by avoiding false multiple minima that would arise from the former charge asymmetry. Similarly, computed CM1, CM3 and Mulliken charges for equivalent atoms in mono- and disubstituted phenyl rings are now symmetrized. E.g., for phenol, the ortho carbons now have the same charge as do the ortho hydrogens, etc. For *m*-cresol, the charges for C4 and C6 are the same as are the charges for their attached hydrogens. The charge symmetrization has been applied to all Z-matrices for molecules in the *mcpro/drugs* directory. These Z-matrices now use the CM1A charges, scaled by 1.14 for neutral molecules [Udier-Blagović, M. et al. *J. Comput. Chem*., **25**, 1322-1332 (2004)]. Improvements which were made to the atom-typing for five-membered heteroaryl rings are also reflected in the new Z-matrices.

5. See NONEBN on page 17. Its use can ensure a correct covalent neighbors list.

6. PDB-format movies can now be written to the SAVE file - see PLTFMT on page 19.

7. 2-Å probe Lennard-Jones particles can now be used as a solvent. This is useful for seeing the shape of a potential protein binding site or empty spaces for a protein-ligand complex. See new test job Probe.

New features in MCPRO 1.67 and 1.68 were:

The first Windows release was for MCPRO 1.68.

The solvent accessible surface area is now decomposed into hydrophobic, hydrophilic (N, O), carbon π (C and aryl H), and weakly polar (halogens, S & P) components. SASA, its components, and the volume for solute ISOLEC are now averaged over the entire MC simulation.

A summary of the key results is provided at the end of the sum file for linear response calculations. All reported quantities are fully averaged over the entire MC simulation. The number of rotor bonds and number of real atoms for solute ISOLEC are now output at the end of the sum file.

The designation of the cap atom in the par file was changed. Instead of an atom number, an atom name, e.g., CAP, is now given. The program then finds the atom with that name in the Z-matrix file. So, the user no longer has to provide the atom number in the par file. See also CAPX.

Significant speed-up was achieved for MC simulations by replacing the ISOLUT function.

New features in earlier MCPRO versions were:

A utility program, *autozmat*, is provided that interconverts many structure file types, e.g., PDB, Sybyl mol2, MDL mol, Gaussian Z-matrix, and BOSS/MCPRO Z-matrix. One can now go from a Sybyl mol2 file to a fully-flexible BOSS/MCPRO Z-matrix in a few minutes. For more information enter *autozmat -h*, when in the mcpro16 directory.

A flat-bottomed harmonic potential has been added for harmonic restraints - Section 10.5. Dihedral angles can also be harmonically restrained - Section 10.9. These additions are useful for refinement of NMR structures using NOE and coupling constant data.

The format for the torsion parameters in the par file has changed. The V0 parameter was removed and four-fold torsions with V4 parameters are now allowed. The phase angles have also been removed. Old par files will still be processed properly; the program looks for the V0 (old style) or V4 (new style) designation in the header at the beginning of the torsion entries. See page 19.

The MAXVAR feature was implemented to allow designation of the maximum number of internal coordinates to be varied on an attempted MC solute move - Section 9.

When NEWZMAT is declared in the par file, the final Z-matrix from an energy minimization or Monte Carlo simulation is appended to the sum file - Section 9.

The intermolecular residue-residue list can be kept constant during an entire MC simulation or in each run by using NORRUP and IFIXRR in the par file - Section 9.

An atom can be restrained to a point in space - see Section 10.5.

New features in MCPRO 1.5 were:

The *solvent accessible surface area and volume* of the solutes are automatically computed. They are determined for the solute structures and printed out at the beginning and end of a run (*i.e.*, not averaged). The solvent probe radius is input in the par file or default values are used. The atomic radii for the solute atoms are calculated from the OPLS σ non-bonded parameters; radius = $2^{1/6}\sigma/2$.

Multiple independent solute or solvent molecules can be used to search for optimal complexes. This can be useful in designing receptors for a guest molecule. (See NOXX and NOSS, page 19.)

All internal-coordinate optimization methods have been improved. A new *conjugate gradient* method (Cartesian-coordinate) has been added (see section 13.2 and Test Job OPT-SH2).

Estimated values for missing bond stretching and angle bending parameters are assigned automatically. Values for bonds are calculated based on atom type and electronegativity (see *Theochem* **312**, 69 (1994)), and angle bending parameters are assigned based on the average of all parameters available for the central atom of the angle.

For users of the BOSS program, the important features in MCPRO versions 1.3 and 1.4 were:

In BOSS, solute sampling is done by choosing a solute at random, performing rigid-body moves, and moving a random subset of the variable bonds, angles, and dihedral angles. In MCPRO, the solute is chosen via the variables NSCHG and NSCHG2 in the parameter file. A residue in that solute is then chosen at random and up to MAXVAR variable bonds, angles, and dihedrals in that residue are varied. A residue-based cutoff procedure is used, replacing the ICUT options in BOSS. Residue designations are given in the Z-matrix atom definitions. Neighbor lists are used for solvent–solvent, residue–residue, and residue–solvent interactions.

The number of possible solutes is 25. Solute entries are separated in the Z-matrix by "TERZ" lines.

5 Operating Systems

MCPRO is normally distributed in *Linux* or *Windows* versions. The *Linux* executables were obtained with the Intel FORTRAN compiler on a 2.66 GHz Intel Core2 quad computer running Fedora Core release 12.0; they should work on any Linux system with a 2.4 or newer kernel.

The *Windows* version of MCPRO runs under *WindowsNT, Windows2000, and WindowsXP* for PC's. A letter to the editor of the *Journal of Computational Chemistry* reporting relative timings for UNIX workstations and Pentium-based PC's is in the August 1996 issue (Vol. 17, p1385-1386). BOSS executes at similar speeds on a SGI R5000 workstation or a 200 MHz Pentium Pro. The *Windows* version provides a powerful molecular modeling system that can also be executed on laptop computers. Display of PDB files of structures output by MCPRO can be performed well on PC's using free software such as *RasMol, VMD, UCSF Chimera, PyMol,* etc.

6 Installation

The MCPRO files are normally provided by ftp. Instructions are given for the installation of the files in each case. For UNIX/Linux, the path to the top MCPRO directory needs to be specified in the *.cshrc* file with a command like
setenv MCPROdir ~/mcpro

The key files are placed in *MCPROdir*. The test jobs and *pepz* utility program files are placed in subdirectories. If only the executable code has been provided, the rest of the manual should be read and the test jobs run. In special cases, the FORTRAN and C source code has been provided in the *source* directory. If compilation is needed, details for UNIX may be found in the file, *Makefile*. The test jobs are then ready to be run. A listing of the names of the distribution files and their contents is provided at the end of this document.

7 Directories and Files

directory	contents
mcpro	main – mcpro executable
mcproman	user's manual and reprints
miscexec	executables for utility programs
scripts	scripts for standard jobs
solbox	solvent boxes
testjobs	test jobs
notes	notes on various topics
pepz	code for *pepz*
chop	code and executable for *chop*
Clu	Code and executables for *clu*
UA	OPLS-UA parameter files
AA	OPLS-AA parameter files
IRIX	IRIX executables (only provided with Unix/Linux)
autozmat189	autozmat source code (only provided with Unix/Linux)
source	MCPRO source code (only provided by special arrangement)
molecules/	
small	Z-matrices for small organic molecules
drugs	Z-matrices for drugs
peptides	Z-matrices for 20 dipeptides

The following files are used by the program. The logical units are assigned in the BLOCK DATA routine and stored in COMMON/IO.

Unit	Suffix	Description
IRD	CMD	The "command file" for executing the program.
ILST	OT	Printed output file (the "out file").
IDSKI	IN	The "in file" contains the coordinates and other data needed to restart the simulation.
IDSKO	UP	The "up file" contains the results for each run that are needed to compute the global averages.
IDSKS	SV	The "save file" is used to store all coordinates periodically for later analysis. The period is specified in the parameter file.
IDSKA	AV	The "av file" contains the averages for distribution functions that may be plotted.
IZM	ZMAT	Contains the Z-matrix input for the solutes.
IPAR	PAR	The "parameter file" contains the potential function parameters and variables for the present simulation.
IBAPAR	oplsua.sb oplsaa.sb	Contains bond stretching and angle bending parameters.
IWAT	watbox	Contains water boxes for initial system setup.
INAQ1	org1box	Contains solvent boxes for non-aqueous solvents.
INAQ2	org2box	Additional solvent boxes for non-aqueous solvents.

IPLT	PLT	The "plt file" receives output coordinates normally in PDB format.
ISUM	SUM	Short version of the printed output file goes to the "sum file".

Most of these files are created automatically by the program. The only user-supplied files are the CMD, PAR and ZMAT files. The IN file is usually empty for an initial system setup. The input for the CMD, PAR and ZMAT files is described below.

MC simulations are carried out in a series of runs, each consisting of *ca.* 100000 to 5000000 (100K – 5M) configurations. A new configuration is generated by randomly moving a solute or solvent molecule or by changing the volume of the system by scaling the coordinates of all molecules. The "in file" is rewritten at the end of each run. The "up" and "save" files grow during the simulation, and the latter can get quite large (this is controlled by NCONSV). If the "up file" is deleted, all averaging information on the simulation is lost.

8 Command File Input

The UNIX command file or *Windows* bat file begins with a header that can be updated. See, for example, the files *lrcmd* and *lr.b*at in Test Job linres. For an MC simulation, a series of runs is typically executed. Each begins with the file assignments. For uniformity, the suffixes for the file names are not changed, though individuals may have their own preferences for designating filenames. The program is executed by the UNIX command, *e.g.*,

```
time ~/mcpro/MCPRO 111 100000 0.15 0.00 0.30
```

or the *Windows* command,

```
%MCPROdir%\mcpro 111 100000 0.15 0.00 0.30
```

The first 3 variables (the `111`) are ICALC, NEWRUN and IPRNT.

ICALC declares the type of calculation:

ICALC = 0	Continuation of a Monte Carlo simulation.
ICALC = 1	Initial set-up for a Monte Carlo simulation. Origins of initial solute and solvent coordinates are declared in the parameter file.
ICALC = 2	Energy minimization for the solute(s) using the Simplex, Powell, Fletcher–Powell or conjugate gradient methods.
ICALC = 7	Perform single-point force field calculation. See the xSPMCP script.
ICALC = 9	Continue MC simulation, but do not update averages - for reequilibration.

NEWRUN determines if averaging is continued or restarted:

NEWRUN = 0	Usual case; continue averaging in a Monte Carlo simulation.
NEWRUN = 1	To begin new averaging by starting a new "up file". NEWRUN is set to 1 for only the first run after equilibration. If NEWRUN is accidentally set to 1 during averaging, the averaging is restarted and the previous runs are lost—though the system is probably now very well equilibrated!

IPRNT determines the verbosity of written output (0–6, least to most):

IPRNT = 0	Usual
IPRNT = 1	Cases.
IPRNT = 2	Also print solute coordinates.
IPRNT = 3	Includes high-energy non-bonded pairs and SASA's for all solute atoms.
IPRNT = 4	Includes residue–residue and solvent–residue lists (use sparingly).
IPRNT = 5	Includes non-bonded pairs list, residues without internal variations, and energy information **for each move** (use VERY sparingly with small MXCON!). For debugging only.
IPRNT = 6	Includes solvent coordinates (ditto).

The next variable is MXCON (the `100000`) which specifies the number of configurations for this run. **Keep it constant during averaging**, *i.e.* after NEWRUN = 1. It is often advisable to perform some very short runs at the beginning for a new system just to check that everything has been set up properly. For example, with a `114` start just run 100 configurations or so (MXCON) and check the system graphically by displaying the resultant plt file. Then restart with `011`, if all is okay. Also, the initial energy may be very high due to some overly short interatomic contacts. These are typically relieved within *ca.* 10K configurations. It is also advisable not to perform volume moves initially (set NVCHG = 999999 in the parameter file), *e.g.*, for the first 300K of equilibration, since undesirable volume expansion may occur to relieve the short contacts.

If ICALC = 2, MXCON has a different meaning. In this case, it is the maximum number of geometry optimization cycles to be performed.

The final three variables (`0.15 0.00 0.30`) are RC0, RC1 and RC2 which define the reference and two perturbed systems for free energy calculations. Normally, they range from 0 to 1, though other values may be convenient for potential-of-mean-force computations. Geometrical and potential function parameters are scaled linearly between initial and final values indicated in the ZMAT file as RC goes from 0 to 1. RC0 is for the reference solute and RC1 and RC2 are for the two perturbed solutes that are treated by the program. Thus, perturbations can be made in two directions (double-wide sampling). If a free energy perturbation is being performed, the parameter and Z-matrix files can be set up so that for new increments (change of RC0) only the command file requires change.

For complete perturbations, the command file needs three sections: equilibration, averaging, and submission of the next job. The later typically includes copying the current in file to be the in file for the next perturbation window. Test Job FEPaq illustrates this. For Linux/Unix only one command file is needed, while for Windows a series of command files is required with one submitting the next. If a free energy calculation is not being performed, set RC0 = RC1 = RC2.

9 Parameter File Input

The parameter file begins with the definition of variables for the simulation. Short descriptions are provided on alternate lines and are mostly self-explanatory. Additional descriptions are in subroutine INIT. Most parameters will be assigned default values, as noted in the file, if a value of zero is specified. Default choices are printed in the ot file for inspection. To obtain a parameter file for a new problem, copy an old parameter file and edit it. A current parameter file is provided as a template in the file *parfile.fomat* in the *notes* directory. Some highlights in the parameter file are:

SVMOD1

Designates the primary solvent choice. The current options from stored boxes are water (TIP3P or TIP4P), methanol, acetonitrile, dimethyl ether, THF, propane, methylenechloride, chloroform, carbon tetrachloride, argon, and DMSO (see also "Available Solvent Boxes", Section 16), or PROBE to use 2-Å diameter LJ spheres - see test job Probe. For GB/SA, designate GBSA. For solventless or non-Monte Carlo simulations, SVMOD1 should be set to NONE.

Solute, Solvent Origin

These two entries designate the origin of the initial **(ICALC = 1)** solute and solvent coordinates. For the solute(s), the options are from the Z-matrix file in Z-matrix (ZMAT), Protein Data Bank (PDB), or MindTool (MIND) format, from the solute coordinates in the in file (IN) (useful if a gas-phase MC run has been performed to get a low-energy solute structure; use the output in file from the gas-phase run as the current in file), and from maximal mapping of the new system onto the one in the in file (ZIN).

With the ZIN option, the solute coordinates come from the in and Z-matrix files. The coordinates of the first 3 atoms of each solute are taken from the in file and used to build the rest of the solute structures with the Z-matrix appropriate for the current RC0 value; the most recent values of the variable dihedrals, angles, and bonds are taken from the in file (see Test Job FEPaq).

For the solvent, the options for the initial (ICALC = 1) coordinates are from the stored boxes (BOXES or CAP), from the coordinates in the supplied in file (IN), or NONE. **When ICALC = 0, by definition, the solute and solvent coordinates come from the in file (IN).**

CAPAT, CAPRAD, CAPFK

When a solvent cap is requested, CAPAT designates the name of the solute atom that defines the center of the cap; we usually call it CAP in the Z-matrix file. CAPRAD gives the cap's radius from CAPAT, and CAPFK is the half-harmonic restoring force constant in kcal/mol–$Å^2$ for solvent that strays beyond CAPRAD. If CAPFK ≠ 0, CAPAT should be placed by itself as a separate solute, and the program will then not attempt to move CAPAT. Otherwise, if CAPAT is a regular solute atom, the solute motion will be constrained by the change in the cap potential when CAPAT is moved. (See also "Cluster Simulations", page 41.) For a single solute in a cap, if CAPAT is designated as CAPX, NROTA1 (below) is used as CAPAT - and set RDELS1 = 0.0.

OPTIMIZER, FTOL

For solute optimizations (ICALC = 2), the choice of methods is Simplex (SIMPLX), Powell (POWELL), Fletcher–Powell (FLEPOW), Steepest Descents (STEEP), or conjugate gradient (CONJUG). FTOL specifies the convergence criterion in kcal/mol between cycles with these optimizers. For ICALC = 2, the remaining control information in the par file is irrelevant except specification of the dielectric constant, SCL14, CUTNB and the plt file format.

NEWZMAT

If NEWZMAT is declared, the final Z-matrix from an optimization or Monte Carlo run is *appended* to the sum file.

NMOL, IBOX, BCUT

IBOX controls the choice of desired solvent box from those listed in Section 16, where IBOX is the number of molecules in the stored box. If the initial (ICALC = 1) solvent coordinates are coming from the IN file, IBOX is automatically set to the number of solvent molecules in the IN file. NMOL designates the total number of solvent molecules to be retained for the present simulation. The program automatically discards the (IBOX – NMOL) solvent molecules with the worst interaction energies with the solute(s). To obtain low initial solute–solvent energies, a rule-of-thumb is to discard an equivalent number of non-hydrogen solvent atoms as there are non-hydrogen solute atoms. For example, if there are 40 such solute atoms and the chosen solvent is chloroform, discard 10 chloroform molecules. For solventless or non-Monte Carlo simulations, both SVMOD1 and SVMOD2 should be specified as NONE and IBOX, NMOL and NMOL2 as 0.

The program will automatically create a solvent box when NMOL = 9999 for ICALC = 1. In this case, set IRECNT = 1 and IZLONG = 1 to center the solute. The box will extend BCUT Å beyond the farthest solute atom in the ± x, ± y, and ± z directions. The disadvantage of this procedure over directly using the stored boxes is that the faces of the box are rough-cut. This leads to a high initial energy, so longer equilibration is required including *ca.* 500K configurations without volume moves (make NVCHG large). However, systems with up to 5000 solvent molecules can be created. Solvent molecules with any atom initially within 2.5 Å of a solute atom are removed. The desired box is cut from a very large solvent cube, *ca.* 100 Å on a side, that is generated from 27 images of smaller, stored boxes.

SVMOD2, NMOL2

If a binary solvent mixture is desired, the second solvent choice and number of molecules are declared by SVMOD2 and NMOL2. The binary solvent boxes are created by first generating a box with NMOL SVMOD1 molecules. NMOL2 of these are replaced randomly by SVMOD2 molecules. Thus, the total number of solvent molecules is NMOL consisting of (NMOL – NMOL2) of type SVMOD1 and NMOL2 of type SVMOD2. The initial total energy for the binary systems will be high, so equilibration for *ca*. 500K configurations without volume moves (set NVCHG = 999999) is recommended. Then,

set NVCHG = 0 to begin the NPT calculations and equilibrate the volume and energy. **Note**: Energy distributions are for the mixture; however, the radial distribution functions are only for atoms in the first solvent. For solventless or non-Monte Carlo simulations, SVMOD1 and SVMOD2 should both be NONE.

NCENT1, NCENT2 The atoms used to define the centers of solutes 1 and 2 for solute centering and translation, as well as for preferential sampling procedures. If NCENT1 = 0, it is reset to 1. If NCENT2 = 0, it is reset to atom 1 of solute 2, if any. The NCENT for other solutes and for solvent molecules is taken as the first atom of the molecule.

NROTA1, NROTA2 The atoms that solutes 1 and 2 are rotated about for the total (rigid-body) rotations. Often, these are set equal to NCENT1 and NCENT2. For free energy perturbations in which geometries are varied, NROTA1 and NROTA2 should be chosen, if possible, to be atoms whose positions are identical in all three solutes' images (the reference and two perturbed ones). Otherwise, the images' relative orientations will change during the simulations. The NROTA for solutes above 2 is taken as the first atom of the solute. If NROTA1 or NROTA2 = 0, they are reset to NCENT1 or NCENT2.

NOBACK NOBACK = 1 should be used if a protein backbone is to be held fixed. RDELS1 and ADELS1 are set to zero to prevent rotation and translation of the protein (solute 1).

NORRUP For MC simulations, if NORRUP = 1, intermolecular interactions will include all residue-residue pairs in the residue-residue list, which is computed at the beginning of each run. If NORRUP = 1, the residue-residue list does not change during a run.

IFIXRR For MC simulations, if IFIXRR = 1, the intermolecular residue-residue list is only computed for ICALC = 1 (at initialization). The same list is then used for all subsequent runs, when ICALC = 0. In this way, a ligand could see the same protein environment for an entire simulation. If IFIXRR = 1, NORRUP is set to 1.

IRECNT To recenter the solute(s) in the middle of the simulation box, set IRECNT = 1. All molecules are translated so the energy is unaffected. This is just cosmetic.

INDSOL Set INDSOL = 1 to let the solutes move independently. If INDSOL = 0, the first two solutes only will be translated in tandem, which is useful for some pmf calculations.

IZLONG Set IZLONG = 1 to have the solute(s) oriented with the longest axis along the z–direction. Only do this at the beginning of a simulation, *e.g.*, when ICALC = 1. (Normally IZLONG = 0 for protein simulations in a solvent cap.)

MAXOVL Set MAXOVL = 1 for FEP calculations to have the perturbed solutes maximally overlaid on the reference solute. The QTRF subroutine then performs least squares fits after every solute move. It can help reduce noise in FEP results and takes little CPU time. (Normally MAXOVL = 0 for protein simulations to avoid global movement of the protein.)

NOXX, NOSS For use with multiple solute or solvent copies. If NOXX = 1, solute-solute interactions are not evaluated except with solute 1. If NOSS = 1, solvent-solvent interactions are not evaluated; only solute-solvent and internal energies are evaluated.

NOBNDV
NOANGV For some equilibrations: if NOBNDV = 1, variable bonds are not varied; if NOANGV = 1, variable angles are not varied.

NONEBN If NONEBN = 1, only bonds declared as variable and additional in the Z-matrix are considered to be covalent bonds for determining covalent neighbors. This is the normal case. Otherwise, bonds are determined by interatomic distance, which may be misleading for a high-energy structure.

NRDF Number of requested solvent–solvent radial distribution functions.

IBDPDB Set IBDPDB = 1 to avoid falsely designating two atoms in different residues to be bonded owing to a bad non-bonded distance between them. Then, in constructing the list of covalent neighbors in routine NBPAIR, MCPRO will only consider atoms in the same residue or N–C or N–CT bonds. For example, a very close contact between salt-bridging ASP and ARG sidechains in a poor x-ray crystal structure might result in bonded sidechain atoms; IBDPDB = 1 rejects these inter-sidechain bonds. Set IBDPDB = 0 if any solute is divided into "residues" not connected by actual peptide bonds. NONEBN = 1 also covers IBDPDB = 1.

STREN Ionic strength in mol/l for GB/SA calculations; see *Ann. Rev. Phys. Chem.* **51**, 129-152 (2000). Default is zero. Has little effect except for nucleic acids.

IDDD To use a distance-dependent dielectric with conjugate gradient minimization or gas-phase MC simulations, set IDDD = 1. Then ε in the potential energy function is DIELEC*r_{ij}. See DIELEC below.

NVCHG,
NSCHG
NSCHG2 An attempt to change the system's volume is made every NVCHG configurations. If an NVT (constant volume) simulation is desired, set NVCHG = 999999. A solute move is attempted every NSCHG configurations for *any* solute and *additionally* every NSCHG2 configurations for solute 2. This is useful for biasing sampling of the ligand (solute 2) in a protein (solute 1)–ligand complex. NVCHG and both NSCHG will be determined automatically, if they are input as 0. For solutes with many internal variables, smaller values of NSCHG are appropriate. After choosing the solute to be moved, one residue in that

solute is randomly chosen for variations in internal coordinates.

NCONROT NCONROT declares the frequency of attempted concerted rotations for the biomolecule's backbone. A conrot move is attempted every NCONROT configurations. The Z-matrix must have been built by *pepz* including a "set conrot" command. See test job FoldProt.

NCONSV Frequency of writing coordinates to the SAVE file for MC runs.

NBUSE Set NBUSE = 1 to use solvent-solvent neighbor lists. For water with fewer than ca. 1000 molecules, this yields no time saving.

MAXVAR Defines the maximum number of variable bond lengths, bond angles, and dihedral angles that will be varied on an attempted solute move (MAXVAR of each type). The default is 15 if MAXVAR = 0. Lower values of MAXVAR increase acceptance rates.

NSAFRQ Frequency for reevaluation of the SA term for MC simulations using GB/SA solvation. Default of zero yields NSAFRQ = 30.

WKC Constant for the preferential sampling using $1/(r^2 + WKC)$ weighting. WKC needs to increase as the number of molecules increases, otherwise there will be gradual volume expansion in NPT simulations. Some typical values of WKC are 150 for 216 or 250 water molecules, and 250 for 267 chloroform molecules. When using a solvent cap, it may be useful to use preferential sampling for initial solvent-only sampling around the solute(s) and to turn it off (WKC = 10000 or other larger number) when the solute is free to move for uniform solvent sampling.

RDEL, ADEL Ranges for translations and rigid rotations of solvent molecules. Program assigns values automatically, when input as zero.

RSOLV Probe radius for solvent-accessible surface calculations.

RDELS1, RDELS2, ADELS1, ADELS2 These are the ranges for attempted translations in Å and rigid-body rotations in degrees for the first and second solutes. *If the ranges are zero, the solutes will not be translated and/or rotated.* The ranges for additional solutes are automatically determined and dynamically adjusted to give *ca.* 50 % acceptance rates for attempted moves.

RCUT The cutoff distance for solvent–solvent interactions based on the NCENTS–NCENTS (usually atom 1–atom 1) distances. For TIP3P and TIP4P water models, this is the oxygen-oxygen distance. All interactions are quadratically feathered to zero between RCUT and RCUT – 0.5 Å (as they are in the BOSS program).

SCUT The cutoff distance for solute–solvent and solute–solute interactions. The latter is feathered as above. Residue-based cutoffs are used. For solute–solute interactions, if any interresidue atom–atom distance is less

than SCUT, the interactions between all pairs of atoms in the two residues are included. For solute–solvent interactions, if any distance between an atom in a residue and solvent atom 1 (O for water) is less than SCUT, the interactions between all atoms in the residue and the solvent molecule are included.

CUTNB The residue-based cutoff distance for intrasolute non-bonded pairs. If any atom in residue X is within CUTNB of any atom in residue Y, all XY atom-atom pairs are included in the non-bonded pairs list. The list is recomputed when a new job (run) is executed.

T, P, TLHT The temperature in °C and the external pressure in atm. The temperature (TLHT) for local heating of any residues so designated in the Z-matrix is given here. The sampling for all other residues and solvent is at temperature T.

DIELEC The dielectric constant, ε, for solute optimizations and continuum simulations. For a distance-dependent dielectric in conjugate gradient optimizations (IDDD = 1, $\varepsilon = DIELEC * r_{ij}$), DIELEC = 4.0 is common. DIELEC is set to 1 for all simulations with explicit solvent molecules.

SCL14C, SCL14L The Coulombic and Lennard–Jones scaling factors for 1,4–intramolecular non-bonded interactions. If both are > 99999, then 1,4 interactions are ignored. The Coulombic and Lennard–Jones 1,4–interaction energies are divided by SCL14C and SCL14L, respectively. The default value is 2.0 for each. This is appropriate for the amino acid torsions included in the *united atom* and *all atom* parameter files and selected automatically by *pepz*.

GSTEP, GSTEP the grid separation in a JAWS calculation, usually 1.0. If it is 0.0 the JAWS procedure is not executed.

GSIZE, GSIZE is the size of the spheres (usually 5.0) that define the grid.

NGSKIP, NGSKIP should be set to 0 in phase 0, 1 otherwise

NGRESTR, NGRESTR restrain theta waters to grid positions. It should be set to 0 in phases 0 and1, and to 1 in phase 2

NTARGET, NTARGET if a specific number of theta waters is desired, otherwise 0

GRIDDIFF, GRIDDIFF, if it is 1 real waters are allowed to diffuse into the grid.

NGRIDATOMS NGRIDATOMS is the number of atoms used to define the grid, if -1 then uses ALL the solute atoms

GRIDSOLATOMS Array containing the indices of atoms used to define the grid for JAWS. This line can be empty if NGRIDATOMS is -1

IFGUP1, IFGUP2 Frequency of update for grid statistics in JAWS phases 1 and 2

PLTFMT The format for coordinates written to the PLT file. The options are MIND, PDB, PDBDM, PDB2, PDBB, and PDBMO. PDB generates a Protein Data Bank format file with dummy atoms removed that is suitable for displaying with normal molecular graphics programs. PDBDM is the same except the (-1) dummy atoms are now also output. PDB2 generates a PDB file including the coordinates for the reference

solute and the first perturbed solute. This is useful for visualizing the structural change for the perturbation. PDBB provides a PDB file with the MCPRO atom types appended after the solute coordinates. Such a file can be read by MCPRO to begin a simulation (page 36). MIND yields a file in the correct format for display with the MindTool program (rarely used). Again, dummy atoms are removed.

PDBMO causes a PDB format movie to be written to the SAVE file with the frequency specified by NCONSV. For perfectly even spacing, make MOD(MXCON, NCONSV) = 0. See test job FoldProt for an example of its use.

ISOLEC The solute–solvent and solute–solute energies are decomposed into their Coulomb and Lennard–Jones components for solute ISOLEC. The default is solute 2 (normally the ligand for protein–ligand complexes). These components are needed for linear response calculations (page 3).

POLSCX Set POLSCX > 0.0 to include residue-residue polarization. POLSCX = 1.0 is the current optimal choice. For no polarization (OPLS-AA), set POLSCX = 0.0 See testjob *polarize* for examples.

The parameter file then lists the non-bonded potential function parameters (q, σ, ε) by atom type and torsional parameters. This is normally done automatically in the command files by appending the *oplsaa.par* or *oplsua.par* files. New atom types should be defined at the end of the current list. Only the parameters that are used for the present calculation need to be listed. *The type numbers do not have to be sequential,* and lines starting with a pound (#) sign are considered to be comment lines. However, since parameters for the stored solvents occur up to atom type 126, the listing of parameters is often not truncated before this point.

The torsional parameters are the Fourier coefficients for different dihedral angle types. Again, new types can be added at the end of the list and only the parameters for the current calculation need to be listed. The torsional potential for angle ϕ is defined as

$$V(\phi) = V1(1+\cos\phi)/2 + V2(1-\cos 2\phi)/2 + V3(1+\cos 3\phi)/2 + V4(1-\cos 4\phi)/2$$

V4 is almost always zero. The input format for specifying the torsional parameters for an angle is:

```
Type   V1      V2      V3      V4       a1-a2-a3-a4

I3  2x F8.4 2x F8.4 2x F8.4 2x F8.4 4x  A2-A2-A2-A2

as in

123xx12345678xx12345678xx12345678xx12345678xxxx

006  1.711     -0.500     0.663     0.0         CT-CT-CT-O?
```

This example is for the CT-CT-CT-OS or CT-CT-CT-OH torsion in alkyl ethers or alcohols. The AMBER atom type quartet at the end of the line is needed for proper automated assignment of dihedral types. The quartet can be entered either as w-x-y-z or z-y-x-w. The ? is a wildcard that is used in the matching algorithm; the priority is exact match > ? > mismatch. Non-bonded (Coulomb plus Lennard-

Jones) interactions are also included within a solute for >1,3 interactions, when there is internal motion.

The complete lists of OPLS united-atom and all-atom non-bonded and torsional parameters are in the files *oplsua.par* and *oplsaa.par*, respectively. These files are normally appended to the top section of the parameter file (see the file *notes/parfile.format*) to yield a complete parameter file. The files containing bond-stretching and angle-bending parameters are *oplsua.sb* and *oplsaa.sb*. *It should be noted here that care must be taken when the parameter files are modified so that the modifications do not conflict with the amino-acid templates (oplsua.db and oplsaa.db) and dihedral-angle assignment files (dihedrals.ua and dihedrals.aa) necessary for the pepz program (Appendix 1).*

10 Z-matrix File Input

Both MC and MD programs utilize as input a representation of the molecular connectivity and geometry known as a Z-matrix, albeit in different fashions. Most MD programs have pre-built Z-matrix libraries for the amino acid residues and some other common small molecules and fragments which are combined, in a step prior to the simulation itself, into the "topology" file which contains only the specification of the bonds, angles, and dihedrals. The coordinates are typically obtained from a PDB file in a separate step.

In MCPRO and BOSS, the Z-matrix is the main input and is utilized by the programs to calculate the energies, positions, and as a "prescription" to generate new conformations. It contains the specifications for the initial setup of the solute geometries, the atom type designations, and the 3-character labels for the atoms that are used in printing. Geometric perturbations are also designated, as well as any internal variables (bond lengths, bond angles, dihedral angles) to be varied. Domain definitions are made for evaluation of intrasolute >1,3 non-bonded interactions when internal variables are sampled over, and excluded atom lists are given (vide infra). Finally, residues designated for sampling with a different temperature, local heating, are listed.

It is important to realize that the Z-matrix is typically used to build the coordinates of the solute molecules when a solute move is attempted. A solute move for a solute with internal variations consists of (1) choosing via NSCHG and NSCHG2 which of the solutes is going to be moved (obviously this step can be skipped if there is only one solute), (2) translating the first three atoms of the solute randomly in all three Cartesian directions, (3) rotating the first three atoms of the solute randomly about one randomly chosen Cartesian axis, and (4) then constructing the rest of the solute from the first three atoms by applying the specifications in the Z-matrix for the remaining atoms of only that solute. Changes in the internal variables are incorporated when the solute is rebuilt from the Z-matrix. If the solute does not have internal variations, steps 2 and 3 are applied to all atoms and step 4 is skipped.

For perturbation calculations, Z-matrices are maintained internally for the two perturbed solute systems (each of which may have multiple fragments). The moves of these perturbed solutes are performed simultaneously with the move of the reference solute system. The same displacements are employed so maximum overlap of the reference and perturbed solutes is maintained. Variations of this procedure are possible by appropriate designations in the parameter file. A handy one is to force the first two solutes to be translated in tandem by setting INDSOL = 0. This is necessary if a potential of mean force is being determined as a function of an intersolute distance. In this case, the two atoms that define the distance need to be declared as NROTA1 and NROTA2. The solutes will then be rotated about these two atoms which will maintain their specified separation that is also maintained by the tandem translation.

Having a good Z-matrix is a very critical component of the simulation process. The creation of a Z-matrix for a protein by hand would be lengthy and prone to error. To automate the process, the program *pepz* (**Pe**ptides and **P**roteins **Z**-matrix Builder) has been developed specifically for the task of creating a Z-matrix in the format read by MCPRO and is discussed further in Appendix 1. When constructing ligand Z-matrices, work from the middle of the molecule outward. This will result in smaller structural changes for variations in dihedral angles and lead to better sampling. Place the first atoms including dummies near the middle of the molecule—do not start the Z-matrix at one end of the molecule.

Note: the Z-matrix should not be modified after the beginning of a Monte Carlo simulation.
The following describes the format of the Z-matrix.

Line 1: `TITLE USED IN PRINTING (A30)`
Subsequent lines:
```
ATOM #I, SYMBOL, TYPE, FINAL TYPE, J, RIJ, K, TH(IJK), L, PHI, RES NAM, RES #
I4 1X A3 1X I4 1X I4 1X 3(I4,F12.6) 1X A3 1X I4

LAST LINE OF Z-MATRIX: BLANK LINE
```

Additional input read by routines ZSTART and NBPAIR is in groupings separated by lines with the first 3 columns blank:

1. Geometry Variations—specify 1 per line (for perturbations):
```
ATOM NUMBER, PARAMETER (1,2, OR 3), FINAL VALUE
I4 I4 F12.6
Then: BLANK LINE
```

2. Bonds to be sampled over—specify the atom number from the Z-matrix, one per line, `I4` or a contiguous range of atoms (`I4-I4`):
```
ATOM NUMBER
I4 or I4-I4
Then: BLANK LINE
```

3. Additional Bonds to include in the energy evaluation, one per line:
```
ATOM 1, ATOM 2
I4 I4
Then: BLANK LINE
```

4. Harmonic Restraints for solute atom pairs, $E = K(R-R0)^2$, one per line (maximum of 90):
```
ATOM I, ATOM J, R0, K0, K1, K2
  I4      I4        4F10.4

or for restraining atom I to a point x, y, z, one per line:
ATOM I, 9999, K, x, y, z
  I4      I4     4F10.4

or for a flat-bottomed harmonic used for NOE restraints, one per line:
ATOM I, ATOM J, -1.0, R-low, R-high, K
I4      I4        4F10.4

Then: BLANK LINE
```

5. Bond Angles to be sampled over—specify the atom number from the Z-matrix, one per line, `I4` or a contiguous range of atoms (`I4-I4`):
```
ATOM NUMBER
I4
Then: BLANK LINE
```

6. Additional Bond Angles to include in the energy evaluation, one per line:
```
ATOM 1, ATOM 2, ATOM 3
I4 I4 I4
```
Or for automatic determination by the program, just declare AUTO in columns 1-4:
AUTO

```
Then: BLANK LINE
```

7. Specify Dihedral Angles to be varied, one per line:

```
ATOM NUMBER, INITIAL TYPE, FINAL TYPE, RANGE
     I4            I4            I4      F12.6
```

Or, for automatic type and range assignments, one can just list the dihedral angles to vary either singly and/or as a contiguous range, e.g., 0005-0014 (the dash is needed):

```
ATOM NUMBER  -  ATOM NUMBER
     I4      -       I4
```

To declare a variable dihedral angle for attempted flipping, anywhere in the listing of variable dihedrals specify (one line for each dihedral to flip)

```
FLIP, RANGE, ATOM NUMBER
flip    I4      I4
```

E.g., to attempt random ±120 degree flips for atom 67, declare
`flip 120  67` (flip can be flip or FLIP)
Typically, RANGE should be 120 or 180. The flips are tried about every 6th attempted MC variation of the angle; the algorithm is in subroutine MOVMOL.

Then: `BLANK LINE`

8. Specify Additional Dihedral Angles to be included in the evaluation of the intramolecular energy, one per line:

```
ATOM 1, ATOM 2, ATOM 3, ATOM 4, INITIAL TYPE, FINAL TYPE
  I4     I4      I4       I4          I4             I4
or
AUTO
```

Then: `BLANK LINE`

Note: for additional bond and dihedral angles, one can mix AUTO with specific designations.

To harmonically restrain dihedral angles, make additional entries:

```
ATOM 1, ATOM 2, ATOM 3, ATOM 4, 500, PHI0   (6I4)
```

Type 500 in the par file is reserved for his purpose. The force constant in kcal/mol-rad^2 is the V4 entry - see oplsaa.par. $E(PHI) = K(PHI-PHI0)^2$.

9. Domain Definitions for exclusion in non-bonded pairs calculations—only if internal variables are sampled over. Enter Domain 1 and Domain 2 pairs, one set per line (4 atoms are needed):

```
1ST ATOM DOM1, LAST ATOM DOM1, 1ST ATOM DOM2, LAST ATOM DOM2
(4I4)
Then: BLANK LINE
```

10. Excluded Atoms List—generally protein backbone atoms for use with NOBACK = 1 in the par file. Also used for designation of residues for concerted rotations. Normally this is generated by *pepz*. Enter excluded atom number, 10 per line:

```
ATOM 1, ATOM 2, ATOM 3, ATOM 4, ATOM 5,... ATOM 10
(10I4)
Then: BLANK LINE
```

11. Local Heating Residues, if desired—specify the residue number from the Z-matrix, one per line, `I4` or a contiguous range of residues (`I4-I4`):

```
RESIDUE NUMBER
I4 or I4-I4
Then: BLANK LINE—TERMINATES
```

12. CG-Frozen Residues, if desired—specify the residue number from the Z-matrix, one per line, `I4` or a contiguous range of residues (`I4-I4`):

```
RESIDUE NUMBER
I4 or I4-I4
Then: BLANK LINE—TERMINATES
```

13. After the "Final BLANK LINE", there can be optional designation of non-bonded parameters for atoms whose charges have normally come from a QM single-point calculation using *BOSS*. For example, if one executes *xAM1SP meoh* with *BOSS*, the output *sum* file is the following:

```
       Methanol                                              Tot. E =    -55.0866
   1 O    800  800    0    0.000000   0    0.000000   0    0.000000         0
   2 DUM   -1    0    1    0.500000   0    0.000000   0    0.000000         0
   3 DUM   -1    0    2    0.500000   1   90.000000   0    0.000000         0
   4 H    801  801    1    0.945698   2   90.000000   3  180.000000         0
   5 C    802  802    1    1.411870   4  108.989752   2  180.000000         0
   6 HC   803  803    5    1.090450   1  110.239001   4  179.999996         0
   7 HC   804  804    5    1.090404   1  110.549523   6  119.850735         0
   8 HC   805  805    5    1.090404   1  110.549526   6  240.149263         0
                    Geometry Variations follow
                    Variable Bonds follow         (I4 or I4-I4)
0004-0008
                    Additional Bonds follow       (2I4)
                    Harmonic Constraints follow
                    Variable Bond Angles follow   (I4 or I4-I4)
0005-0008
                    Additional Bond Angles follow (3I4)
AUTO
                    Variable Dihedrals follow     (3I4,F12.6)
0006-0008
                    Additional Dihedrals follow   (6I4)
AUTO
                    Domain Definitions follow     (4I4)
                    Conformational Search         (2I4, 2F12.6)
                    Final blank line

 Final Non-Bonded Parameters for QM (AM1 CM1Ax1.14) Atoms:

 800  8 OH  -0.586957  3.120000  0.170000
 801  1 HO   0.406869  0.000000  0.000000
 802  6 CT  -0.043638  3.500000  0.066000
 803  1 HC   0.074575  2.500000  0.030000
 804  1 HC   0.074575  2.500000  0.030000
 805  1 HC   0.074575  2.500000  0.030000
```

Note that the usual OPLS-AA atom types have been replaced by types 800-805, whose non-bonded parameters including the 1.14*CM1A charges have been designated at the end of the new Z-matrix. Atom type numbers 800-899 and 9500-9999 have been reserved for this purpose, and the corresponding non-bonded parameters are read from the Z-matrix file rather that the *oplsaa.par* file. The Z-matrices in the *drugs* directory all have this form.

Some explanatory details:

10.1 Atom Input

The first atom is placed at the origin (0,0,0), the second atom is on the +x axis and the third atom is in the xy plane. (All of the rectangular solvent boxes have the long axis as z.) TYPE and FINAL TYPE specify the initial (RC0 = 0) and final atom types (RC0 = 1) for atom i. The corresponding non-bonded potential function parameters are retrieved from the parameter file by the PARAM subroutine. If FINAL TYPE = TYPE, then FINAL TYPE can be left blank or shown the same as TYPE.

The order in which atoms are defined will affect sampling. Refer to pages 31–33 for implications of dihedral angle and residue definitions within the body of the Z-matrix. Usually for a protein/ligand system, the protein would be defined first (solute 1) and the ligand would follow (solute 2). In the cases investigated to date, CAPAT is defined alone as solute 3.

***** Very important ***:** For systems with multiple solutes, separate them with a line having "TERZ" in columns 1–4. Also note there are two types of dummy atoms; one with type –1 is a true dummy that is only used in constructing the solute coordinates from the Z-matrix, while a dummy with type 100 has $q = \sigma = \varepsilon = 0$, but can be converted in a perturbation calculation to a real atom with $q \neq \sigma \neq \varepsilon \neq 0$. Initial type –1 dummy atoms in the Z-matrix should be given residue number 0 (e.g., see Test Job COX2). Solvent caps may be centered on dummy atoms of either type, however, only type 100 atoms will appear in plt files written during the simulation. It may be convenient to place the cap center near but *not coincident with* a solute atom.

10.2 Geometry Variations

This designates how geometrical parameters are to be changed for a perturbation. The ATOM NUMBER corresponds to the numbering in the Z-matrix input including dummy atoms. PARAMETER refers to bond length (1), bond angle (2), or dihedral angle (3). The initial parameter value is in the Z-matrix and refers to RC (or λ) = 0. The final parameter value is specified here and refers to RC = 1. The parameter value is scaled linearly for intermediate values of RC:

$$\theta(RC) = RC \times (\theta_{final} - \theta_{initial}) + \theta_{initial}$$

Note: in the command file, the RC values can be outside the (0,1) range. For example, in computing a potential of mean force as a function of a distance, one could state in the Z-matrix file that the initial distance is 0 Å for RC = 0 and 1 Å for RC = 1. Then, if RC = 4, for example, in the command file, the program knows to set the system up with the distance as 4(1-0) + 0 = 4 Å. However, for molecular mutations, *e.g.*, $CH_3OH \rightarrow CH_3CH_3$, one just lets the mutation run from RC = 0 to 1 since the potential function parameters need to start as methanol and end up as ethane as RC goes from 0 to 1.

10.3 Bond Length Variations

Bond lengths in the Z-matrix can be varied during the Monte Carlo sampling by specifying the atom that is attached by the bond. Specify one atom per line in `I4` format or a contiguous range of atoms in `I4-I4` format (include the hyphen). The harmonic bond stretching parameters are retrieved automatically from the files oplsua.sb or oplsaa.sb. Any missing bond-stretching parameters are flagged in the output file; values will be estimated based on atom type and electronegativity. If the atom types

for the bond are changing during a free energy calculation, the parameters for the bond in the perturbed solutes are also determined automatically. The ranges for the bond length variations are determined by the program. Only the variable bonds in the chosen residue are varied on each attempted solute move (see subroutine MOVMOL for details).

10.4 Additional Bonds

These are bonds that are not explicit in the Z-matrix, but that are to be included in the energy evaluation. The typical case is a ring closure bond, such as occurs in proline residues. Specify the two atoms in the bond (`2I4` format), one pair per line. Again, the parameters are retrieved automatically.

10.5 Harmonic Restraints

Bond Restraint. Harmonic restraints can be included between any pairs of solute atoms (maximum of 20 pairs). For the I-J pair, RIJ0 specifies the reference separation in Å, and K0, K1 and K2 are the force constants in kcal/mol-Å^2 for this restraint in the reference solute and the two perturbed solutes, if any. Each restraint causes the following energy term to be added to the total energy where K = K0, K1 and K2 for the three solutes:

$$EBC = K(RIJ - RIJ0)^2$$

EBC is added to the solute-solute energy, EXX. Even if there is only one solute, the restraints energy is included in the total energy.

The restraints are useful in some special contexts. For example, in mutating one host/guest complex A···B to another C···D, one could perturb from A···B to A—B to C—D to C···D where ··· means the two molecules are completely free, while the introduction of the harmonic constraints is symbolized by —. This keeps the solutes from floating apart during the intermediate stages of the mutation. We used this procedure for mutating the Watson-Crick GC complex to AT in chloroform using two constraints with the first step corresponding to G···C (K0 = 0) going to G—C (K1 = 10, R0(GN1-CN3) = R0(GO6-CN4) = 3.0). G—C was then mutated to A—T with K0 = K1 = K2 = 10, and in the last step (A—T to A···T), the restraints were removed. Other uses, such as restrained optimizations are possible.

Position Restraint. Atoms can also be restrained to a fixed point in Cartesian space. This can be useful for FEP calculations in which a solute is being annihilated - see J. Hermans & L. Wang, *J. Am. Chem. Soc.* **119**, 2707 (1997) and N. McDonald et al., *J. Am. Chem. Soc.* **120**, 5104 (1998). In this case the format for the entry in the Z-matrix file specifies the chosen atom (I), atom J is 9999, and then the force constant and the x, y, and z coordinates for the fixed point are specified. The contribution from each such restraint is then given by EBC = KR^2, where R is the distance between atom I and the fixed point. IRECNT is automatically set to 0, independent of its value in the parameter file, so that the solutes are not automatically recentered at the end of each run, as this would shift atom I relative to the fixed point.

Alternatively, atom J can be restrained to the average of the positions of 1, 2, or 3 other atoms. In this case, specify atom I = 9999, atom J is the atom to be restrained, RIJ0 is the force constant, and K0, K1, K2 are the numbers for the three atoms (these are specified as floating point, e.g., 23.0 for atom 23, and converted to integers by the program).

NOE Restraint. A flat-bottomed harmonic can also be applied between atoms I and J:

$$EBC = K(RIJ - Rhigh)^2 \qquad \text{for } RIJ > Rhigh$$

$$EBC = 0 \quad \text{for } Rhigh > RIJ > Rlow$$

$$EBC = K(RIJ - Rlow)^2 \quad \text{for } RIJ < Rlow$$

10.6 Bond Angle Variations

Bond angles in the Z-matrix are varied by specifying the atom for which the angle appears in the Z-matrix. The angle-bending parameters for the harmonic bend are retrieved and the range for attempted variations of the angle is computed automatically. Any missing parameters are flagged in the output file; estimated values are based on the average of all parameters available in the stretch-bend file for the central atom(type) of the angle. Only the variable angles in the chosen residue are varied on each attempted solute move.

10.7 Additional Bond Angles

As for bonds, additional bond angles can be included in the energy evaluation. Specify the three atoms in order x–y–z where y is the central atom, one triple per line. ***Note**: The first two bonds and intervening angle for each solute can not be variable. Start with two –1 dummies, if this is a problem.***

The program will automatically determine the additional bond angles by designating AUTO in columns 1-4. In this case, central atoms in explicitly declared variable and additional bond angles and atoms in additional bonds are processed such that all bond angles about these atoms will be included in the MM energy evaluation. The angles that have been added are listed in the ot files.

10.8 Dihedral Angle Variations

For dihedral angles that are sampled over (*i.e.*, varied during the simulation), specify the atom set up by the dihedral angle in the Z-matrix and the dihedral angle INITIAL TYPE number whose corresponding Fourier coefficients must be available in the parameter file. INITIAL TYPE is the initial value for RC0 = 0 and FINAL TYPE is the type for RC0 = 1.

If the INITIAL TYPE is 0, the torsional potential is taken as 0, though the angle is still varied (useful for solute optimization). Often, FINAL TYPE = INITIAL TYPE, and in this case FINAL TYPE can be set equal to INITIAL TYPE or 0. However, if an atom in the dihedral angle changes type, the dihedral type may change, too. If a torsional potential of zero is desired for FINAL TYPE, a new torsion type must be specifically defined in the parameter file with V's = 0 (torsion type 100). If both INITIAL and FINAL TYPE are 0, the torsional potential is taken as 0, though the angle is still varied (useful for solute optimization).

Also specify the RANGE (PDEL) for variation of the dihedral angle in degrees. Attempted variations of the dihedral angle will be made within ± RANGE degrees of the current value of the dihedral angle. If the PDEL for a dihedral angle is set to zero, the angle will not be explicitly varied but its value will be recorded in the output and average files for later analysis.

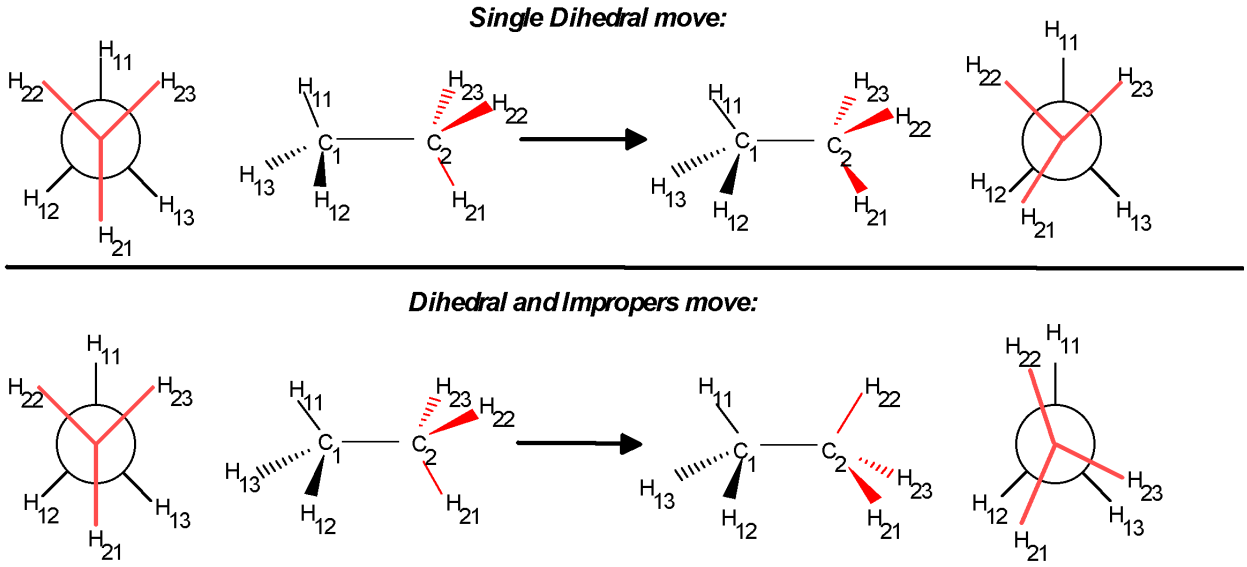
***New in MCPRO 1.65**:* Alternatively, the program will automatically assign the dihedral types and ranges by designating a group of dihedrals to vary, e.g., 0004-0025, or for a single angle designate,

e.g., 0004-0004. The - is needed as the key that auto assignment is desired. Specifically assigned dihedrals and auto ones can be mixed. The assignments are made from the corresponding quartet of AMBER atom types. If there are missing parameters or multiple matching quartets, they are noted in the ot file along with the program's processing decision. The TYPE entries are irrelevant, while if a range PDEL for a single entry is specifed, it is used. The default ranges are currently ±2 degrees for improper dihedral angles or ±12 degrees otherwise.

Options for a single angle:

0004-0004	program determines type and range for angle 4
0004 -1 -1 10.0	program determines type and uses ±10° for the range
000400230023 10.0	full specification by user for angle 4 as type 23 with range ±10°

Improper dihedrals may be defined in the atom input section of the Z-matrix to ease sampling of rotation around single bonds. The utility of this technique can be seen in the rotation around the $–C_1–C_2–$ bond in ethane as shown in the scheme below. The top portion illustrates a typical motion done by a change in the dihedral angle $H_{21}–C_2–C_1–H_{11}$ when the remaining hydrogens are defined by normal dihedrals ($H_{22}–C_2–C_1–H_{11}$, $H_{23}–C_2–C_1–H_{11}$, etc.) As shown, the motion of H_{21} changes its bond angles to H_{22} and H_{23} rather dramatically, with a large increase in the energy and almost surely the rejection of the motion by the Metropolis algorithm. A smaller range of motion will be needed which will decrease the efficiency of sampling. The bottom part of the scheme shows the effect of the same change in the $H_{21}–C_2–C_1–H_{11}$ dihedral when both H_{22} and H_{23} are defined using improper angles ($H_{22}–C_2–C_1–H_{21}$, $H_{23}–C_2–C_1–H_{21}$, etc.). All the bond angles around C_2 are kept constant, and the entire methyl group rotates as a rigid body. This motion is energetically more economical and has a better probability of being accepted. To complete the sampling for H_{22} and H_{23}, small variations (ca. 3°) in the dihedrals $H_{22}–C_2–C_1–H_{21}$ and $H_{23}–C_2–C_1–H_{21}$ can be specified. Many examples can be found in the molecules/small directory, e.g., see ethane.z. See also page 64 for an example of the use of improper dihedral angle definitions to enforce stereochemistry in united atom models.

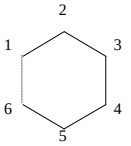


***** Very important ***:** In cases where sampling includes dihedral angle variations, the dihedral changes are made assuming the first 3 atoms of each solute can be used to build the rest of the solute from the Z-matrix. Do not use atoms from any solute to define other than the first three atoms of another

solute; the dihedral angles for the first three atoms of any solute can not be varied except for solute optimizations (ICALC = 2). Use of dummy atoms can solve any problems.

10.9 Additional Dihedral Angles

These are dihedral angles that are to be included in the evaluation of the intramolecular energy, but that are not defined (or varied) explicitly in the Z-matrix. Such angles occur in flexible rings, for example. For the carbon framework of cyclohexane, only 3 dihedral angles would be necessary to define the positions of the atoms in the body of the Z-matrix, but all 6 should be included in the energy evaluation. The 3 additionals can be listed or use AUTO.



Dihedrals explicit in the body of a Z-matrix which would be declared variable as in 10.8 and assigned appropriate potentials:

```
4-3-2-1, 5-4-3-2, 6-5-4-3
```

Additional dihedrals to be included in the energy evaluation:

```
1-6-5-4, 2-1-6-5, 3-2-1-6
```

A dihedral angle phi can also be harmonically restrained by declaring it as an additional dihedral, specifying the initial type as 500 and the final type as the reference value for phi, phi0, as an integer in degrees (0-360). This is useful for NMR structure refinements. V4 for entry 500 gives the force constant in kcal/mol-rad^2. E(phi) = K(phi-phi0)^2.

10.10 Domain Definitions

The domain definitions cause intrasolute non-bonded interactions **not** to be evaluated between any atoms in domains 1 and 2. A domain may be a single atom or atoms occurring sequentially in the Z-matrix. Domain 1 can equal domain 2. If there are torsions, all other **intra**solute non-bonded interactions will be included and listed in the ot file. Subroutine NBPAIR can be consulted for details. For example, with united-atom t-butyl alcohol if the torsion for the H on O is included (type 54 V_3 = 0.65 kcal/mol) defined as C–C–O–H, this Fourier term fully specifies the torsional potential. However, non-bonded interactions with the H may be included. These can be excluded by the appropriate domain designations.

Domains are also used in PHE, TYR, HIS, TRP, and ARG amino acid side-chains when built using *pepz* (Appendix 1). All bonds, angles and dihedrals within the domains are fixed to maintain ring or guanidinium planarity, and the relative position of the atoms then remains the same. The energy from the interatomic interactions is therefore a constant, which makes its evaluation unnecessary for a MC simulation. Check the non-bonded pairs list at the beginning of the ot file (printed when IPRINT = 5) to verify that the proper exclusions are being made.

10.11 Excluded Atom Lists

The excluded atom list is similar to the domain definition in that intrasolute non-bonded interactions are **not** evaluated between any pairs of atoms in the list. However, atoms in this list need not be contiguous. It may be used for atoms which are frozen throughout the simulation, for example, a rigid protein backbone.

10.12 Local Heating Residues

Residues may be listed here if "local heating" is desired to increase barrier crossing in a MC simulation. The temperature given in the parameter file, TLHT, will be used for these residues in the Metropolis test, effectively increasing the acceptance of these moves. This may be useful for early stages of equilibration.

10.13 CG-Frozen Residues

Residues may be listed here if it is desired to freeze them during a conjugate gradients or steepest descent optimization. Their energy contributions are still calculated, but their movement is stopped by setting their derivatives to zero. Note that these lines are not needed for most calculations unless this feature is desired. A utility program *fixzm22* is included in the *miscexec* directory to add the proper header line to Z-matrix files from previous MCPRO versions.

10.14 Residue Definitions

Attempted moves are not made for fixed residues. The program automatically determines this when there are no variable bonds, angles, or dihedrals for the residue.

The residue-based sampling and cutoff procedures used in MCPRO were implemented to improve sampling for proteins, which naturally are made up of residues, by not moving large portions of the system at once, avoiding high rejection rates. However, it may be advantageous to divide small peptide and organic solutes into residues as well. This will often speed up a calculation, as fewer interactions are calculated for each move of the solute; only the interactions associated with the residue that varies would be updated. Also, by not moving all variable bonds, angles, and dihedrals in a given solute at once (as would happen for a solute not broken into residues), larger dihedral angle variations (PDELS) may be used in the Z-matrix to obtain similar or better acceptance ratios. **Note:** residues should be neutral or have integer charges to avoid errors in electrostatic energies due to the residue-based cutoffs. The output file lists the charge of each residue.

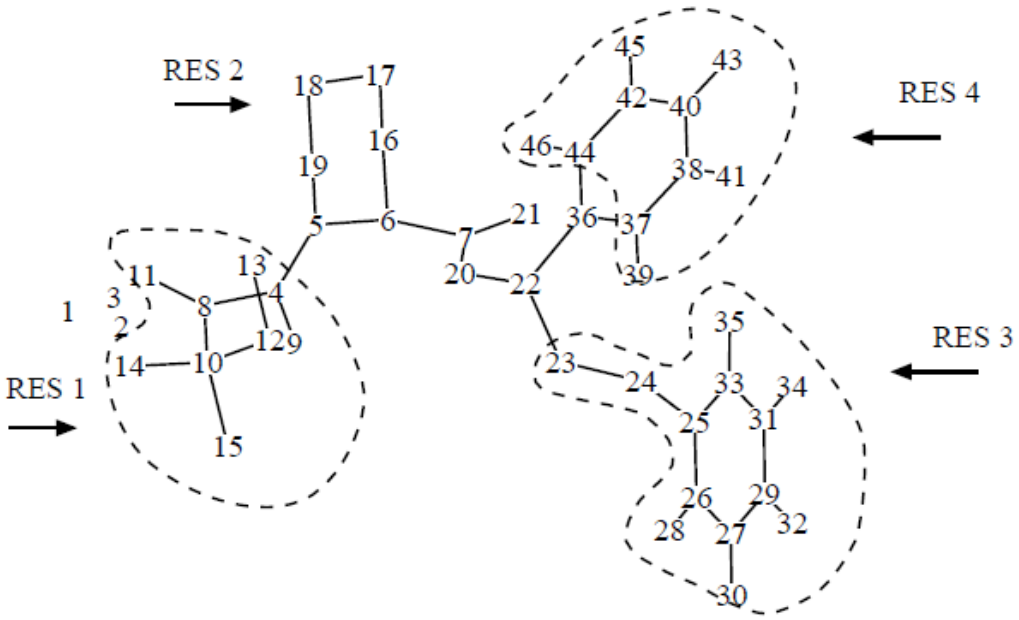
Peptides are mentioned above with organic solutes because it is often useful to build a ligand Z-matrix from the center of the molecule outward. This will ensure that dihedral angle variations of the main chain of a flexible ligand near the beginning of the Z-matrix (N-terminus if instead built sequentially) do not "drag" the remainder of the molecule with them, causing the moves in that region to be frequently rejected. For peptide and organic ligand Z-matrices built manually, residues may be defined in any order, as long as:

1. IBDPDB = 0,

2. the first real atom of the solute is in residue #1 of that solute, and

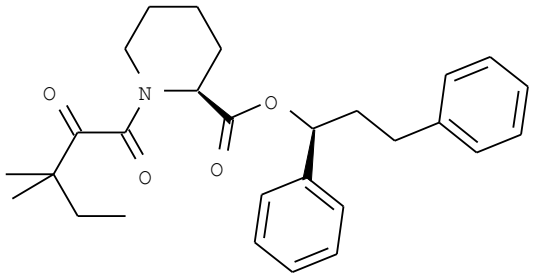
3. the last atom of the solute is in the last residue of that solute.

The sample Z-matrix shown in section 10.14 meets the criteria; the structure is pictured below. It may be compared to a similar structure given in Test Job FEP-FK, which was built as a single "residue" when IBDPDB = 1 in the parameter file prevented alternative residue definitions.



10.15 Sample Z-matrix

The following is an example of a complete Z-matrix file for a mutation with one solute, "SKF10" and an atom (CAPAT) to anchor a solvent cluster:




```
Column numbers:
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456
```

```
FKBP inhibitor SKF10 (RC0=0) & SKF5 (RC0=1) for FEP
   1  Du   -1   -1    0   0.000000    0    0.00000    0    0.00000  UNK    0
   2  Du   -1   -1    1   1.000000    0    0.00000    0    0.00000  UNK    0
   3  Du   -1   -1    2   1.000000    1   90.00000    0    0.00000  UNK    0
   4  C8  415  415    3   3.093105    2  106.12040    1  169.97566  S10    1
   5  N7    3    3    4   1.351647    3  119.49152    2 -109.97965  S10    2
   6  C2  416  416    5   1.457917    4  119.90911    3 -172.92623  S10    2
   7  C1   58   58    6   1.541875    5  118.15725    4  -91.76426  S10    2
   8  C9  157  157    4   1.544688    5  124.11036    6  173.07400  S10    1
   9  O3    2    2    4   1.232652    8  117.56871    5 -175.90999  S10    1
  10 C10  417  417    8   1.536100    4  121.84116    5 -112.70472  S10    1
  11  O4  158  158    8   1.236564    4  118.30376   10 -174.67685  S10    1
  12 C11   71   71   10   1.518039    8  110.58396    4   59.51818  S10    1
  13 C12   68   68   12   1.543865   10  114.01691    8   68.24818  S10    1
  14 C13   70   70   10   1.544923    8  112.29335   12  119.77535  S10    1
  15 C14   70   70   10   1.571590    8  112.94848   12 -118.58937  S10    1
  16  C3    9    9    6   1.525374    5  102.15712    7 -125.76412  S10    2
  17  C4    9    9   16   1.523358    6  115.85396    5   61.16179  S10    2
  18  C5    9    9   17   1.515852   16  108.22384    6  -60.82073  S10    2
  19  C6   15   15   18   1.530162   17  106.12126   16   54.08324  S10    2
  20  O1   62   62    7   1.317372    6  115.81155    5   16.96428  S10    3
  21  O2   59   59    7   1.202703    6  122.14702   20 -175.69681  S10    2
  22 C15  418   61   20   1.438753    7  122.25948    6 -171.82594  S10    2
```

```
  23 C16   71   71   22   1.531943   20  106.93905    7  -93.45815  S10    3
  24 C17  304  304   23   1.527380   22  117.86595   20  167.99840  S10    3
  25 C18  230  230   24   1.517087   23  114.19512   22 -139.32103  S10    3
  26 C19  230  230   25   1.322131   24  120.14060   23  -91.98138  S10    3
  27 C20  230  230   26   1.317171   25  120.45912   24  166.97797  S10    3
  28 H19  231  231   26   1.090000   25  120.00000   27  180.00000  S10    3
  29 C21  230  230   27   1.327057   26  120.38556   25    0.43611  S10    3
  30 H20  231  231   27   1.090000   26  120.00000   29  180.00000  S10    3
  31 C22  230  230   29   1.319366   27  119.66210   26    0.47711  S10    3
  32 H21  231  231   29   1.090000   27  120.00000   31  180.00000  S10    3
  33 C23  230  230   31   1.328055   29  119.56540   27   -0.35866  S10    3
  34 H22  231  231   31   1.090000   29  120.00000   33  180.00000  S10    3
  35 H23  231  231   33   1.090000   25  120.00000   31  180.00000  S10    3
  36 C24  230  100   22   1.546330   20  106.81247   23  122.16958  S10    2
  37 C25  230  100   36   1.335268   22  118.43611   20  -89.59100  S10    4
  38 C26  230  100   37   1.320975   36  121.14124   22 -177.08140  S10    4
  39 H25  231  100   37   1.090000   36  120.00000   38  180.00000  S10    4
  40 C27  230  100   38   1.324688   37  119.63093   36   -0.26281  S10    4
  41 H26  231  100   38   1.090000   37  120.00000   40  180.00000  S10    4
  42 C28  230  100   40   1.327607   38  120.22091   37    0.04046  S10    4
  43 H27  231  100   40   1.090000   38  120.00000   42  180.00000  S10    4
  44 C29  230  100   42   1.327375   40  120.26177   38   -0.02243  S10    4
  45 H28  231  100   42   1.090000   40  120.00000   44  180.00000  S10    4
  46 H29  231  100   44   1.090000   36  120.00000   42  180.00000  S10    4
TERZ
  47 CAP  100  100    3   7.083       2   80.2        1  153.3      COC    5
                    Geometry Variations follow    (2I4,F12.6)
  36   1   0.65
  37   1   0.35
  38   1   0.35
  39   1   0.35
  40   1   0.35
  41   1   0.35
  42   1   0.35
  43   1   0.35
  44   1   0.35
  45   1   0.35
  46   1   0.35
                    Variable Bonds follow         (I4 or I4-I4)
                    Additional Bonds follow       (2I4)
   5  19
                    Harmonic Constraints follow   (2I4,4F10.4)
                    Variable Bond Angles follow   (I4 or I4-I4)
   6-0025
  36
                    Additional Bond Angles follow (3I4)
   9   4   5
  19   5   4
  19   5   6
  16   6   7
  21   7  20
  10   8  11
  14  10  12
  15  10  12
  14  10  15
  18  19   5
  36  22  23
                    Variable Dihedrals follow     (3I4,F12.6)
   7 047 047   1.000
   8 127 127   1.000
  10 128 128   1.000
  12 130 130   1.000
  13 132 132   2.000
  17 058 058   1.000
  18 073 073   1.000
  19 073 073   1.000
  20 049 049   1.000
  22 126 126   1.000
  23 135 136   1.000
  24 058 073   1.000
  25 073 073   1.000
  26 060 060   1.000
  37 134 100   1.000
                    Additional Dihedrals follow   (6I4)
   8   4   5  19 127 127
   9   4   5   6 127 127
   9   4   5  19 127 127
   7   6   5  19 076 076
  16   6   5   4 048 048
  16   6   5  19 078 078
```

```
 20   7   6  16 050 050
 11   8   4   5 129 129
 10   8   4   9 129 129
 11   8   4   9 128 128
 12  10   8  11 131 131
 15  10   8   4 130 130
 15  10   8  11 131 131
 14  10   8   4 130 130
 14  10   8  11 131 131
 13  12  10  14 132 132
 13  12  10  15 132 132
 17  16   6   7 058 058
  5  19  18  17 073 073
 18  19   5   4 079 079
 18  19   5   6 080 080
 36  22  20   7 135 135
 36  22  23  24 058 100
 33  25  24  23 060 060
 37  36  22  23 133 133
 44  36  22  20 134 100
 44  36  22  23 133 100
                  Domain Definitions follow     (4I4)
 24  35  24  35
 36  46  36  46
 36  46  22  22
                  Excluded Atoms List follows      (10I4)
                  Local Heating Residues follow (I4 or I4-I4)
                  CG-Frozen Residues            (I4 or I4-I4)
                  Final blank line
```

In this example, the phenyl substituent of a FKBP inhibitor is removed by converting it into dummy atoms (type 100). Atom types and dihedral angles to this substituent are appropriately modified. Dihedral type 100 is a dummy dihedral type with V's = 0.0. Bond lengths are fixed, and the rigid phenyl rings are defined as domains to simplify the calculation of non-bonded interactions, but otherwise the molecule is flexible. The solvent cap is centered on atom 47, also a type 100 dummy atom. It is a separate solute (note the TERZ line), is defined with respect to the initial type –1 dummy atoms, and will not be moved in MCPRO. Type 100 was chosen rather than -1 so that the atom would appear in PDB files for display purposes. It was placed initially near atom 22 which was the atom closest to the center of the inhibitor. The molecule is broken up into residues as described previously for improved sampling and computational efficiency. The residue name is read by MCPRO but not used; for display purposes, the entire inhibitor was named "S10". See also Test Job FEP-FK.

This example does not have harmonic constraints or excluded atoms; however, it helps avoid mistakes to include the shown information on the formally blank lines. In creating a new Z-matrix file, it is often useful to just edit an old one to get the formats right. The *pepz* program is certainly helpful for building new protein Z-matrices, and the XChemEdit and BOMB programs (not included in this distribution) developed in the Jorgensen lab is useful for building ligand Z-matrices. These may then be used with *pepz* or may be modified by hand for use in MCPRO. Most user mistakes are made in the Z-matrix file. When running long MCPRO jobs, display the plt files often to check the solute structures; this can reveal user errors.

11 PDB Input

The MCPRO program can read a PDB (Protein Data Bank) file directly to input the coordinates for the solute(s). Solvent can then be added, as usual, from the BOXES or IN files. The solute is treated as a rigid entity in this case—**no intramolecular degrees of freedom are varied**. Nevertheless, such simulations can be useful for equilibrating the solvent around a molecule and for studying the molecule's solvation. Test Job PDB provides a short example of this type.

The PDB input file is used in place of the Z-matrix file and needs to begin with a one line title (`6X,A30`). Subsequently, only ATOM and HETATM entries are recognized. The atom name and coordinates are extracted from these records. TER can be used to separate molecules, but SSBOND, REMARK, etc. records are ignored. The atom input is terminated by a blank line, not a TER. This needs to be followed by the MCPRO atom types for the ATOM and HETATM entries in order in `10I3` format.

Free energy calculations can be performed for varying the atom types and/or shrinking the solute as in an annihilation to obtain the absolute free energy of solvation. After the initial atom types, there is one line with the variables IFREE and ISHRNK in `2I1` format.

IFREE = 1 to read subsequent final atom types.
ISHRNK = 1 to scale the solute coordinates as specified by RC0, RC1 and RC2.

If IFREE = 1, the final atom types in `10I3` format are then specified. Otherwise, the IFREE, ISHRNK line can be left blank to end the PDB input file. For an annihilation, charge stripping before shrinking is recommended (*Tetrahedron* **47**, 2491 (1991)). In the stripping, the true atom types are perturbed to a set with the same Lennard–Jones parameters, but zero charges. Then, in the shrinking, the chargeless set is perturbed to type 100 dummy atoms and the solute is simultaneously contracted. The input PDB file is not written over, however, it is needed for each run to supply the atom types. As usual, the running coordinates for the MC simulation are kept in the in file. A PDB file for the current solute coordinates can be written to the plt file by setting PLTFMT to PDB or PDBB in the parameter file.

12 Coordinate Input in Mind Format and Reaction Path Following

The program is able to read a file containing a sequence of structures that are perturbed between in free energy calculations. This was used in the Jorgensen group to study reactions of small molecules in solution for which a "movie" has been made using the reaction path following procedure in GAUSSIAN 92. See the following papers for applications to Diels–Alder and Claisen reactions: *J. Am. Chem. Soc.*, **113**, 7430 (1991); **114**, 10966 (1992). The structures that are generated along the minimum energy reaction path are placed sequentially in one file, which replaces the usual MCPRO Z-matrix file. MCPRO will then perturb between the structure entries that are requested using RC0, RC1, and RC2 to identify the reference and two perturbed structures. This then allows calculation of the change in free energy of solvation along the reaction path. The structures in the file are treated as one solute and are held rigid during the simulations. There is automatic processing in MCPRO that (1) does a least–squares fit to maximally overlay the reference and two perturbed solutes, and (2) adds a dummy solute atom as the last atom in each structure. It is placed near the structures' centers and has the same coordinates in each structure. This dummy atom is set to be NROTA1, so the rigid-body rotations for the structures are made about this point.

For example, to begin a job to perturb the 2nd structure in the file to the 1st and 3rd, one specifies in the command file:

```
time $mcpro 111 $configurations 2.0 1.0 3.0
```

ICALC is handled as usual, `1` for set-up, `0` to continue. MIND needs to be specified in the parameter file as the solute format, and it remains as MIND for all subsequent runs. The solvent origin can either

be BOXES or IN. The latter uses the solvent coordinates in the in file from a previous run; it is often advantageous in this case to run the calculation for *ca.* 5000 configurations with only solute moves (NSCHG = 1) to help position the solute well in the solvent hole. See Test Job Mind for an example calculation for the rotational barrier of acetamide.

The format for the MindTool coordinate file is given below. Free format is used and a pound sign (#) in column 1 for any line causes the line to be ignored, *e.g.*, a comment line. A percent sign (%) marks the end of a given structure's coordinates. Note that the partial charge and Lennard–Jones parameters have to be specified for each atom. For example,

```
#Title line with a 30 character title beginning in column 21 (1–20 blank).
#12345678901234567890
                     THIS IS THE TITLE LINE
# coordinate lines for structure 1:
atomic number (integer), x, y, z, q, σ, ε (real), atom symbol (optional)
atomic number (integer), x, y, z, q, σ, ε (real), atom symbol (optional)
atomic number (integer), x, y, z, q, σ, ε (real), atom symbol (optional)
%
#Coordinate lines for structure 2
atomic number (integer), x, y, z, q, σ, ε (real), atom symbol (optional)
…
```

13 Solventless Calculations

13.1 Continuum and GB/SA Simulations

The program can also be used to perform Monte Carlo simulations for the solute(s) in the absence of solvent. This is useful for obtaining gas-phase or continuum model reference data. In the parameter file, the solvent designation needs to be blank (SVMOD1 = SVMOD2 = NONE) and the dielectric constant (ε) for the medium should be specified where indicated—the table below may be useful.

Solvent	T (°C)	ε	bp (°C)
vacuum		1.0	
argon	–191	1.5	–185.9
water	25	78.3	100.0
methanol	25	32.66	64.5
ethanol	25	24.55	78.3
formamide	20	111.0	210.5
NMA	32	191.3	206.7
acetic acid	20	6.17	117.9
acetonitrile	25	35.94	81.6
DMSO	25	46.45	189.0
dichloromethane	25	8.93	39.6
chloroform	20	4.81	61.2
carbon tetrachloride	25	2.23	76.6
dimethyl ether	25	5.02	–24.8
diethyl ether	25	4.20	34.4
THF	25	7.58	66.0
benzene	25	2.27	80.1

The dielectric constant is used to scale all Coulombic interactions. The only other change is in the parameter file where the solvent box size IBOX and NMOL need to be specified as zero.

The program will then perform NVT calculations as instructed in which every attempted move will be for the solute(s). A ligand can be moved in the environment of a fixed protein with gas phase MC if the protein is solute 1, NSCHG = 999999, and the ligand is solute 2, NSCHG2 = 1. The output only contains the thermodynamic results and dihedral angle distributions. Note that vacuum calculations could be used to search for conformational minima of a single solute or minima for solute pairs. High T runs could generate many starting structures which could then be quenched in low T simulations. As usual, use ICALC = 1 to begin the simulation and ICALC = 0 to continue it. See test job MCgas for an example of a standard MC simulation of a single molecule in the gas phase.

Alternatively, **GB/SA** solvation can be invoked by declaring SVMOD1 = GBSA. The total energy for the sampling is then the gas-phase energy plus the free energy of hydration using the GB/SA model. The GB/SA treatment is based on the model described in Qiu, D.; Shenkin, P. S.; Hollinger, F. P.; Still, W. C., *J. Phys. Chem. A*, **101**, 3005-3014 (1997). A principal difference is that OPLS-AA sigmas divided by two (σ/2) are used as the van der Waals radii, R_{vdW}, whereas Still et al. used OPLS-UA sigmas and a united-atom model for CH_n groups. Improved results are obtained with our OPLS-AA implementation; a paper summarizing results is in the *MCPROman* directory.

13.2 Energy Minimizations

Direct energy minimizations for the solute(s) are performed via ICALC = 2 and choosing an optimization method in the parameter file (page 17). SVMOD1 should be NONE or GBSA, and SVMOD2 should be NONE in the parameter file. The bond lengths, bond angles, and dihedral angles that are declared as variable in the Z-matrix are optimized.

Non-Gradient Methods

- Downhill SIMPLEX
- POWELL's direction set method

Gradient Methods

- Davidon-Fletcher-Powell (FLEPOW) method
- Conjugate Gradient method (CONJUG)
- Steepest Descents (STEEP)

Both the non-gradient and gradient methods are designed to find a local minimum near the starting geometry. The gradient methods assume that the objective function is quadratic, and therefore they are not likely to be efficient and may result in a wrong solution, if the geometry is far off the minimum. A good starting geometry speeds convergence, and some care may be needed to avoid having bond angles pass through 180°, which can make the dihedral angle poorly defined. In this case, the Z-matrix must be rewritten in such a way that the linearity problem does not occur at any bond angles throughout the optimization; dummy atoms may be helpful.

For internal coordinate optimization (SIMPLEX, POWELL, FLEPOW) of two solutes, the intersolute geometry normally requires 6 variables to be optimized. It is often convenient to use r, θ, and φ for the first atom of the second solute, θ and φ for the second atom, and φ for the third. The program knows to set the corresponding force constants to zero for any such variable bond lengths, bond angles and dihedral angles that involve atoms in two solutes; declare the type for any such dihedral angle as zero. These six variable declarations should be removed from the Z–matrix before Monte Carlo simulations are run. The program also knows to make force constants involving –1 dummy atoms zero, so dummies can be used in defining the optimized variables. **Note:** the *pepz* program will NOT automatically include these intersolute variables (normally connecting real ligand atoms to initial –1 dummy atoms in the ligand or in the protein). The user is responsible for including the additional degrees of freedom in the Z-matrix for these optimizers.

It may be worthwhile to try different optimizers to see if they converge to the same or different minima. If an optimization exceeds the maximum number of iterations, it can be restarted from the termination point. To do this, declare NEWZMAT in the parameter file, which causes the final Z-matrix to be written to the sum file. Delete the old Z-matrix file, copy the sum file to be the new Z-matrix file, and submit the job for another optimization cycle. It is recommended to make at least one such resubmission to verify convergence, particularly for relatively flat potential energy surfaces. Recall that FTOL (page 17) controls the convergence criterion. More information on the optimization methods can be found in subroutines SIMPLEX, POWELL, FLEPOW, and CONGRD.

Conjugate gradient and **steepest descents** minimizations operate in Cartesian space rather than internal coordinates. They are much faster than the other methods. Intersolute degrees of freedom do not need

to be explicitly included in the Z-matrix; however, **the solute to be optimized must be defined as fully flexible** in terms of bonds, angles, and dihedrals between real atoms. With the current version of the *pepz* program, this is only possible for proteins with the **all-atom force field and *pepz* database files**. (The united-atom implementation assumes improper torsions will be fixed, to maintain chirality at α-carbons and planarity for amide bonds, etc. This may be updated in the future with the addition of high torsional barriers for improper torsions, as are used in the all-atom force field.) With the all-atom parameters, however, the commands to set variable all, set variable unsaturated, set variable impropers, and set override domains must all be used in pepz to obtain a fully-flexible Z-matrix. For more details, see Appendix 1 and Test Job Thrombin. An additional option for CONJUG and STEEP minimization is the use of a distance-dependent dielectric model, using the flag IDDD =1. If needed, some residues can be kept at their initial positions by including them in the CG-Frozen residues list, the last line in the Z-matrix before the QM-derived parameters.

If the maximum number of variables that can be optimized (NMAX) is exceeded, a warning will be issued and the optimization will stop. Test Job OPT-UA demonstrates an optimization of the united atom force field (all-atom aromatic) phenylalanine dipeptide with the Fletcher-Powell method. Test job thrombin is a conjugate gradient minimization of a protein-ligand complex with the all-atom force field, and both constant and distance-dependent dielectric options are employed (separately). Pepz input is also demonstrated in each case.

14 Cluster Simulations

Monte Carlo simulations can be performed for a solvent cluster with or without solutes. Periodic boundary conditions are <u>not</u> used. Since the volume is not well-defined, only NVT calculations are performed (no volume moves, NVCHG = 999999). The system is initially set up by designating in the parameter file the central solute atom for the cluster (CAPAT; it may be a dummy atom), the initial radius of the cluster (CAPRAD in Å), and the restoring force constant (CAPFK in kcal/mol–Å^2). A half-harmonic potential is applied to keep solvent molecules from drifting away, if desired. The solute–solvent energies (ESX) are incremented by CAPPOT, where R is the distance from atom 1 of the solvent molecule to CAPAT:

$$CAPPOT = 0;\ R \leq CAPRAD$$
$$CAPPOT = CAPFK(R - CAPRAD)^2;\ R > CAPRAD$$

For simulation of the true cluster, select CAPFK to be zero. In other applications, a weak restoring potential, e.g., CAPFK = 0.5, may be desirable. If CAPAT is a dummy atom alone as a separate solute, the program will not move it, allowing the cap center to be stationary.

With the solvent origin as BOXES, the program automatically generates the solvent coordinates from the stored boxes, and discards any beyond CAPRAD or within 2.5 Å of a solute atom. NMOL, IBOX, and BCUT in the parameter file are ignored; the program will determine NMOL based on the choices of CAPAT and CAPRAD. To discard solvent molecules with bad interactions, (1) make the cap with the desired CAPRAD by running just a few configurations, (2) see how many solvent molecules were generated, and (3) restart (ICALC = 1) with the solvent origin defined as IN and make NMOL less than the number of molecules in the cap.

With the solvent origin as IN, the program will read the initial solvent coordinates from the in file and center the cap on CAPAT. If NMOL is less than the number of solvent molecules in the in file, the remaining solvent molecules with the energetically worst interactions with the solute are discarded. These molecules (and the next five for good measure) have their interaction energies listed in the output file. Files for several equilibrated water caps are provided, *e.g.*, wcapin.22 is a water sphere with 22 Å radius. These files can be used to obtain initial solvent coordinates for simulations of solutes in a water cap; for ICALC = 1, the solute and solvent origins would be ZMAT and IN. The 22 Å cap has 1503 water molecules; the 1503 – NMOL waters with the worst interactions with the solute are automatically discarded. The solvent model may be chosen as either TIP3P or TIP4P and will be created accordingly from these in files. The value of NMOL can be modified until the solvent molecules that are discarded begin to have low energies of interaction with the solute.

15 Pure Liquid Simulations

Monte Carlo simulations in the NVT and NPT ensembles can be performed for the eleven pure solvents that have been provided. A Z-matrix file is needed with a single dummy atom of type –1, e.g.,

```
Liquid Water Simulation
0001 DUM   -1
(then 12 blank lines)
```

The solvent name needs to be specified at the top of the parameter file and NSCHG can be set very large to prevent moving the dummy particle. Make WKC large, *e.g*., 10,000. The atom type of –1 causes the dummy atom to be ignored in all energy calculations. See the test job MCwater. Simulations for other solvents can be performed with the BOSS program, but not with MCPRO.

16 Available Solvent Boxes

Solvent	NMOL	Dimensions (Å)	References for Potential Functions
Water	216	18.6 X 18.6 X 18.6	*J. Chem. Phys.* **79**, 926 (1983)
	250	17.1 X 17.1 X 25.6	*Mol. Phys*. **56**, 1381 (1985)
	267	20.0 X 20.0 X 20.0	
	324	18.6 X 18.6 X 27.9	
	400	20.0 X 20.0 X 30.0	
	512	25.0 X 25.0 X 25.0	
	750	25.0 X 25.0 X 37.5	
Methanol	128	20.9 X 20.9 X 20.9	*J. Phys. Chem.* **90**, 1276 (1986)
	192	20.9 X 20.9 X 31.3	
	267	26.7 X 26.7 X 26.7	
	400	26.7 X 26.7 X 40.0	
Acetonitrile	128	22.6 X 22.6 X 22.6	*Mol. Phys*. **63**, 547 (1988)
	192	22.6 X 22.6 X 33.9	
	267	28.9 X 28.9 X 28.9	
	400	28.9 X 28.9 X 43.3	
Dimethylether	125	23.8 X 23.8 X 23.8	*J. Comput. Chem*. **11**, 958 (1990)
	267	30.3 X 30.3 X 30.3	
	400	30.3 X 30.3 X 45.4	
Propane	128	25.7 X 25.7 X 25.7	*J. Am. Chem. Soc.* **106**, 6638 (1984)
	192	25.7 X 25.7 X 38.6	
	267	33.0 X 33.0 X 33.0	
	400	33.0 X 33.0 X 49.5	
Chloroform	128	25.7 X 25.7 X 25.7	*J. Phys. Chem.* **94**, 1683 (1990)
	192	25.7 X 25.7 X 38.6	
	267	33.0 X 33.0 X 33.0	
	400	33.0 X 33.0 X 49.5	
Methylenechloride	128	24.3 X 24.3 X 24.3	*J. Am. Chem. Soc.* **116**, 3494 (1994)
	192	24.3 X 24.3 X 36.4	
	267	30.4 X 30.4 X 30.4	
	400	30.4 X 30.4 X 45.6	
Tetrahydrofuran	128	25.8 X 25.8 X 25.8	*J. Comput. Chem*. **11**, 958 (1990)
	192	25.8 X 25.8 X 38.7	
	267	32.5 X 32.5 X 32.5	

	400	32.5 X 32.5 X 48.8	
Argon	128	18.5 X 18.5 X 18.5	*Mol. Phys*. **24**, 1013 (1972)
	192	18.5 X 18.5 X 27.8	
	267	23.0 X 23.0 X 23.0	
	400	23.0 X 23.0 X 34.5	
Carbon Tetrachloride	128	27.6 X 27.6 X 27.6	*J. Am. Chem. Soc.* **114**, 7535 (1992)
	192	27.6 X 27.6 X 41.4	
	267	34.6 X 34.6 X 34.6	
	400	34.6 X 34.6 X 51.9	
Dimethyl Sulfoxide	128	24.5 X 24.5 X 24.5	Unpublished
	192	24.5 X 24.5 X 36.8	
	267	30.7 X 30.7 X 30.7	
	400	30.7 X 30.7 X 46.0	

The boxes have been equilibrated at 25 °C except for dimethyl ether (–24.84 °C), propane (–42.07 °C), and argon (–185.85 °C) which were equilibrated at their normal (1 atm) boiling points.

The length of the calculations is sensitive to the number of sites in the solvent molecules. For N sites, N^2 intersolvent distances must be calculated to evaluate a solvent–solvent interaction energy. To a first approximation, the length of the calculations is, consequently, proportional to N^2, *i.e.*, the computer time required for a simulation in THF is 25/9 = 2.8 times that for a simulation in methanol. United atom CH_n groups are used for all stored solvents.

17 Aqueous Solvent Setup - JAWS

The default procedure to solvate a protein in MCPRO overlaps a sphere of water molecules onto a protein structure and eliminates water molecules that show steric clashes. The procedure may occasionally miss hydration sites in buried pockets or trap water molecules into unfavorable pockets. Inadequate positioning of water molecules has been shown to decrease the accuracy of MC/FEP simulations (see Michel, J.; Tirado-Rives, J.; Jorgensen, W. L., *J. Am. Chem. Soc*. **200**9, *131*, 15403-15411). The procedure JAWS has been developed to overcome these difficulties as detailed in Michel, J. Tirado-Rives, J. Jorgensen, W. L., *J. Phys. Chem. B* **2009**, *113*, 13337-13346.

JAWS is typically used to optimize the placement of water molecules in the vicinity of a bound ligand. The user to defines regions of space in the binding site where hydration sites will be sought. The parameters needed to define it are set in the par file. This region is represented by a rectangular grid of separation GSTEP, bound by atom-centered spheres of radius GSIZE. Spheres of radius GSIZE on a NGRIDATOMS whose z-matrix indices are listed below the line "Array of Grid Atoms". The set of overlapped spheres define the volume used to equilibrate water molecules. Alternatively, one can set NGRIDATOMS to -1 and the volume will be defined by placing a sphere on each ligand atom. A complete JAWS calculation has three distinct phases. Phase 0 is usually a short solvent equilibration and initialization of the grid. Phase 1 is used to detect the plausible hydration sites, and phase 2 evaluates their affinities (see test job JAWS). Once the run is completed the JAWS-derived solvent distribution is available in the output in-file JAWS-all.in. Because this in-file also contains the coordinates of the protein-ligand complex, it is usually convenient to create a solvent in-file that contains the coordinates of the solvent only for use with other protein-ligand Z-matrices. This can be done using the utility makesolventin.py.

If the JAWS-derived water distribution is used with another protein-ligand z-matrix, there can be steric clashes between water and protein/ligand atoms, it is thus advised to perform a rapid solvent

equilibration before a production run. To initiate an MCPRO simulation with a z-matrix and a solvent in-file, set in the par file the solute keyword to ZMAT and the solvent keyword to IN. NMOL must also match the number of water molecules present in the solvent in-file (the first number listed in the solvent file). See the test jobs in the folder JAWS for details.

18 Test Jobs

Job	Name	Description
1	Ala6	Prepare & optimize Ac-(Ala)6-NHMe in extended and alpha-helical forms.
2	A6-box	Solvate Ac-(Ala)6-NHMe in a water box and run a short MC simulation.
3	A6-cap	Solvate Ac-(Ala)6-NHMe in a water droplet and run a short MC simulation.
4	OPT-UA	Build & optimize Ac-Phe-NHMe with the older OPLS-UA force field.
5	linres	Linear response calculation for ibuprofen in a water box.
6	avibio	MC for avidin/biotin complex in a water box with solvent only sampling.
7	cydex	MC for beta-cyclodextrin in a water box.
8	COX2	COX2 with inhibitor system ready to run linear response.
9	OPT-SH2	Conjugate Gradient optimization for a complex with Src-SH2 domain.
10	CDK2	Instructions for setup of CDK2 protein/ligand complex from raw PDB file.
11	FEPaq	Simple FEP calculation - acetaminophen para-hydroxy -> para-chloro in water box.
12	FEPgas	Simple FEP calculation - acetaminophen para-hydroxy -> para-chloro in the gas phase.
13	FEPphi	Simple FEP calculations - compute free energy profile for conversion of acetaminophen trans amide -> cis amide, and a psi scan for Ala dipeptide.
14	FEP-FK	MC/FEP calculation for an FKBP/inhibitor complex in a water cap; UA force field.
15	FEP-SH2	MC/FEP calculation for a complex with src-SH2 domain in a water cap.
16	HIVP	MC for HIV protease with SB203238 inhibitor in a water cap.
17	thrombin	linear response MC calculations for the thrombin/NAPAP complex in a water cap.
18	PDB	example of rarely used PDB input format; HIV protease in a water cap.
19	Mind	example of rarely used Mind input format; FEP for amide torsion.
20	MCgas	MC for a single molecule in the gas phase. Illustrates Flip option.
21	dAdCdT	Build and optimize dAdCdT with the OPLS-AA force field.
22	157D	Build from PDB entry 157d.ent and optimize a double-helical RNA dodecanucleotide.
23	MCGBSA	MC with GB/SA solvation for (Ala)6
24	MCwater	MC simulation for pure TIP4P water at 25 C and 1 atm.
25	FoldProt	MC simulations to fold the TrpZip2 peptide using concerted rotations.
26	FoldRNA	MC simulations of an RNA tetraloop using concerted rotations.
27	Probe	Saturate a protein binding site with 2-Å diameter probe spheres.
28	FEP-RT	Current setup for protein-ligand FEP calculations. HIV-RT is the used.
29	polarize	Illustrates calculations with inclusion of residue-residue polarization.
30	FEP-ScanCl	Illustrates a Cl to H FEP scan for a protein-ligand complex.
31	SOSgas	Standard FEP calculation using overlap sampling in the gas phase.
32	SOSaq	Standard FEP calculation using overlap sampling in water.
33	SOS-Prot	Overlap sampling FEP calculations for protein-ligand binding.
34	JAWS	Illustration of aqueous solvent setup using the JAWS procedure.
35	HalogenBond	Illustrates the treatment of Cl, Br, I with X sites.
36	OPT-RT	Illustration of a Conjugate Gradients optimization with frozen residues.
37	MM-GBSA	MM-GB/SA calculation of the difference on free energy of binding of UC-10 and UC-781 to wt-HIV RT.

More than thirty test jobs have been provided. The command files for these are ready to be executed. All that is ever needed to begin work on a new problem is the linked program, and the command or bat, parameter, Z-matrix and, sometimes, in files. For the typical initial set-up from the Z-matrix and stored solvent boxes, the in file is initially empty. The files with the solvent boxes, *watbox*, *org1box* and *org2box*, and the files with stretching and bending parameters, *oplsua.sb* and *oplsaa.sb*, must also be

accessible as specified in the *cmd* or *.bat* files. All other files are created automatically by the program. Many of the test Monte Carlo runs that are to be executed are short, *i.e.*, just for instruction. Also, view the generated plt files to see the systems in detail **Note**: Due to arithmetic differences, MC results may vary slightly from computer to computer. Technically, the Markov chain diverges as soon as one configuration is accepted/rejected on one computer and not on the other.

Each test job has a README file with the key information on the test job. Also, the subdirectories *molecules/small* and *peptides* contain hundreds of OPLS-AA Z-matrices for organic molecules and dipeptides that are ready for optimization and that can be used to help construct Z-matrices for other molecules. Use of *pepz* is the easiest way to build biomolecules. Additional information is provided below on some of the test jobs.

Test Job 5 - linres
An example of an MC simulation for a single solute, ibuprofen, in water is provided. The results are used to predict properties of the drug. Please see the README file for details.

Test Job 6 - avibio
An example of an MC simulation for avidin/biotin in a large water box (N = 4781) with water only MC sampling. Please see the README file for details.

Test Job 9 – OPT-SH2
The *src*-SH2 domain complexed with a Parke-Davis inhibitor (experimental data from *Bioorg. Med. Chem.* **5**, 41 (1997)) is included to demonstrate *pepz* input for an all-atom, fully-flexible system for conjugate gradient optimization. The input PDB file, fixed.pdb, is derived from the crystal and NMR structures (one loop) of this SH2 domain with a similar phosphopeptide (pdb1shd.ent and pdb1hcs.ent, respectively). Full flexibility is the default in the current version of *pepz*, but the new commands "set parameter type all" and "set override domains" are required to specify all-atom residue-templates will be used and to prevent the removal of degrees of freedom associated with domain definitions. Note that these commands must precede the sequence definition or reading of the PDB for sequence information. Inhibitor atoms in inhzmat were given unique temporary names for *pepz*, and the final Z-matrix optzmat contains the original, more intuitive names for the atoms. The positions of HB for all valine residues and HG for all leucine residues were checked and manually repositioned if they had been placed collinearly with CG2 or CD2 (*i.e.*, dihedral values in Z-matrix for VAL HB and CG2 were both set to 120.0, while defined by the same three atoms). See Appendix 1 for details on why this might have occurred. Poor positioning of these atoms may be resolved by optimization, but one should use the best starting geometry possible with conjugate gradient methods. The dihedral angles for the β-hydrogen of VAL 34, 38, 59, 87, and 104 were changed to –120.0 degrees in optzmat; all leucine γ-hydrogens were placed correctly.

The xOPTCG and xOPTCGDD scripts are used to run two separate optimizations of the protein-ligand complex using the same initial conformation from optzmat. The first is a constant-dielectric conjugate-gradient optimization, which uses a dielectric constant of 1. Next, a distance-dependent dielectric of $4.0*r_{ij}$ is used with the xOPTCGDD script. In each optimization, 50 is the maximum number of cycles to be run (MXCON). With a system this large, the goal is to remove initial bad contacts, not run to strict convergence. However, a job that does not run to completion may be resubmitted, if desired. Comments in the output file will state the reason the optimization ended and whether or not resubmission is recommended. This job takes about 45 min on an SGI R5000.

Test Job 14 – FEP-FK

This is an example of a free energy perturbation window for the disappearance of the phenyl substituent of an inhibitor of FK506 binding protein (FKBP) in a 22 Å cap of TIP4P water molecules. The entire protein from the crystal structure file (pdb1fkg.ent) was used, with residues further than 12 Å from the parent compound FK506 held fixed. We have named the inhibitor SKF8 or SK8, and the analogous inhibitor without the phenyl is SKF5 (*J. Am. Chem. Soc*. **115**, 9925 (1993)). The inhibitor was defined from the nitrogen atom outwards, and has not been divided into residues.

The parameter file specifies the TIP4P water model, a 22 Å cap, ZIN/IN for the solute/solvent origin, and NVCHG = 999999, NSCHG = 10, NSCHG2 = 24. The simulation is carried out at 25 °C, and no preferential sampling is performed (WKC = 10000) so that water molecules at the edge of the cap get an equal opportunity to equilibrate. IBDPDB = 1 is turned on (IBDPDB = 0 was not an option when this simulation was set up).

The window begins with copying the in file from the previous window, in this case equilibration of the complex at RC0 = 0.000. *ZIN/IN as the solute/solvent origins in the parameter file will allow the solutes to be rebuilt at each* `111` *start. For example, the solute must be rebuilt on coming from the previous window for equilibration to reflect the new reference configuration.* Equilibration of the complex in the new window is performed with RC0 = RC1 = RC2 = 0.050 for computational efficiency. Five runs of MXCON = 1000 are performed (10 runs of MXCON = 200000 in actual simulation) for equilibration in this window. *Then, ZIN/IN and a fresh* `111` *start allows the structures to be rebuilt for the averaging simulation, with RC0 ≠ RC1 ≠ RC2, to compute the free energy associated with the perturbed configurations.* For the FEP portion, RC0 remains 0.050, but RC1 = 0.025 and RC2 = 0.075, and another set of runs is performed. The final lines of the command file are commented out, but they demonstrate how the next window could be started from this file. The in file is copied into a new directory for the next window, the command file for the next window is executed, and the command file for this window then exits:

```
cp skfin ../10/skfin
cd ../10
csh -e skfcmd >>& skflog &
```

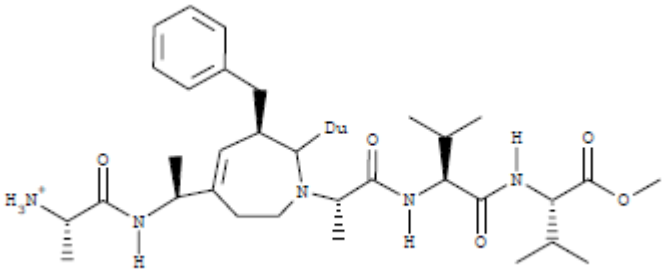
See skf.bat for the analogous submission of batch files on PC's. On a Silicon Graphics R5000 machine, this test simulation takes 15 minutes.

Note: The sample Z-matrix of section 10.14 is for a perturbation of SKF10 to SKF5. This molecule differs from SKF8/SKF5 in the stereochemistry about atom 36 (C24) and has been divided into residues for use with IBDPDB = 0 in the parameter file.

Test Job 15 – FEP-SH2

This test job illustrates a complete FEP calculation for perturbing amide **1** to methylamide **2** bound to Src Sh2 domain, as reported in Bioorg. Med. Chem. Lett. **2000**, *10*, 2067. See the README file.

Test Job 16 – HIVP



In this example, the HIV–1 protease complexed with the above SB203238 inhibitor (*J. Med. Chem.* **38**, 3246 (1995)) will be built for an amide to amine perturbation of the inhibitor. The united-atom force field (all-atom aromatics) is used for both protein and ligand. A Monte Carlo simulation of the complex will then be run in a 22 Å TIP3P water cap. The origin of the coordinates of the complex is the Brookhaven PDB file, pdb1hbv.ent. The Z-matrix for the amide inhibitor was built separately from the amine form of the crystal structure and is included in the file, inhzmat. The system was "chopped" down from the original 200 residues to include only residues with an atom less than 12 Å from the inhibitor, "GAN". The residues to be removed were determined graphically and were removed manually from the PDB file. Backbone atoms from the residues just before and after the remaining "first" and "last" residue were included to position ACE and AME capping groups (file: modified.pdb). The chain designations for each chain of the dimer were removed and the remaining residues were renumbered from 1 using the *fixpdb* utility program included. It was desired to protonate one of the catalytic ASP residues of the protease, and this residue was renamed ASH in the PDB file to correspond to the *pepz* database (oplsua.db or oplsaa.db) name of a protonated ASP residue. The resulting PDB file for use with *pepz* is called fixed.pdb.

The input file for *pepz*, pepz.in, is also included. When the program is executed, ACE and AME capping groups are added as they appear in the sequence, the backbone of the protein is frozen (consistent with only certain residues of the protease remaining in the simulation system), side-chains further than 9 Å from the inhibitor in the crystal structure are frozen, and the protein is centered.

Note: The protonated ASP residue is listed as ASH in the sequence, and in other systems histidine protonation states should be dealt with similarly. The inhibitor and a solvent cap anchor atom are added via the "read boss" Z-matrix command. Atom names in the PDB file correspond exactly to those in the Z-matrix of the inhibitor, inhzmat. A cap atom is included in the Z-matrix, and its coordinates are also given in the PDB file (nearly the same coordinates as an inhibitor atom). The inhibitor Z-matrix residues were numbered so that there is no residue zero, even for initial dummy atoms (see Appendix 2). The command

```
xPEPZ pepz
```

results in pepz.zmat The variable and additional bonds, angles, and dihedrals from the inhibitor may be incorporated in a different order than they were listed in the inhibitor Z-matrix, and all atoms from the inhibitor Z-matrix are directly included in the new complex. A PDB file with the three initial dummy atoms, pepz.pdb.out, is written out for a graphical comparison with the originating PDB structure. The file pepz.out is written and demonstrates that all residues were found and hydrogens and capping groups were added where necessary.

The file pepz.zmat was changed in the following ways to obtain the final version, hivzmat: (1) dihedral angle variations for the protein were changed for higher acceptance ratios in sampling based on the suggestions included on page 65, (2) the inhibitor was divided into residues as suggested on page 28, and (3) variable bond angle designations were collected and sorted from the *pepz* (and MCPRO version 1.3) `I4` format to the shorter `I4-I4` range format.

The MC simulation is started with hivcmd. or hiv.bat. IBDPDB = 0 is used to allow non-standard residue designations for the inhibitor, which was broken up into 5 residues and built from the central core of the molecule outward. Preferential sampling is used to bias sampling of water molecules near the solutes by setting WKC = 50. The simulation is performed at 37 °C to be consistent with the temperature at which the inhibitor binding constants were measured.

MXCON = 10000 in hivcmd for 5 MC runs (*ca.* 1 minute each on a SGI R5000 machine). In actual use, 10 runs of 500K configurations were used for full equilibration.

Test Job 17 –Thrombin

This example demonstrates MC simulations of the thrombin inhibitor NAPAP in solution and bound to thrombin for use in linear response calculations. The system structure is based on the Brookhaven PDB file pdb1ets.ent, and both protein and ligand are modeled with the all atom force field. After conjugate gradient minimization of the crystal structure the complex was manually truncated 16 Å from the inhibitor with only those residues within 10 Å of the inhibitor being sampled. Each system is solvated with a 22 Å cap of TIP4P water and simulated at 25 ºC.

Since the inhibitor is charged, special care was taken to insure that its environment within the protein is neutral. In addition to neutralizing all charged residues near the system boundary which might not be well solvated, two additional negatively charged residues were neutralized to give the protein a net charge of zero. Furthermore, the solute-solute cutoff in the parameter file and the residue-residue list in the "in" file were adjusted in the bound simulation so that each residue of the inhibitor interacts with all charged residues of the protein. The bound simulation uses IFIXRR =1, so the residue-residue list is constant for all simulations.

The ligand and protein-ligand systems are separated into the directories 'unbound' and 'bound', respectively, with each Z-matrix setup in the 'pepz' subdirectory. The inhibitor molecule is built by PEPZ from the residues NAS, GLI, PAP, and PIP in the PEPZ database file test9.db. Due to some unusual features of the inhibitor residues, some variable and additional angles and torsions in the Z-matrices produced by PEPZ were changed or eliminated to generate the final Z-matrices.

For this example 50000 steps of water-only equilibration have been run previously to generate preeqin, which is copied in the command file to thrnain or napin in the bound and unbound simulations, respectively. The equilibration and averaging are run from this configuration in 5 blocks of MXCON = 10000. Under normal circumstances the water-only equilibration might consist of 5 runs of MXCON = 200000 and the full equilibration and averaging 20 runs each of MXCON = 500000.

Test Job 18 – PDB

This is an example of PDB input format (rarely used), using a portion of HIV protease as described in hiv.pdb. The pdbcmd file executes a 10K-configuration MC simulation for 1000 water molecules about the rigid protein. A solvent cap is centered on atom 1, a dummy atom at the center of the protein. The

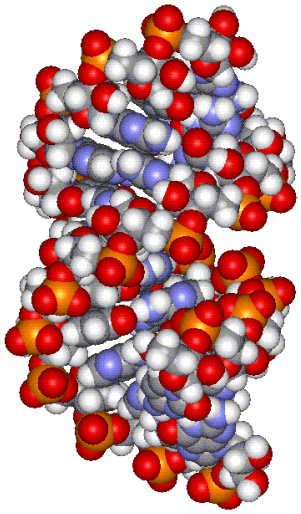
solvent cap is read from the in-file for an equilibrated water droplet with 22-Å radius, wcapin.22. Though the cap contains 1503 water molecules (this can be seen on the first line of the wcapin.22 file), designation of NMOL as 1000 in the pdbpar file causes the 1000 water molecules with the most favorable interactions with the protein to be retained. The PDB input file contains the usual atom and residue information and Cartesian coordinates, followed by the MCPRO atom types for each atom. The final line with IFREE = 0 and ISHRINK = 0 indicates that no perturbation is performed; this option is only realistic for small solutes.

Test Job 19 – Mind

Use of the MIND input format (rarely used except previously for studying reactions in solution) is illustrated with the rotation about the amide bond of united-atom acetamide. The acetcmd file is designed for rotation in 30° increments, and the second structure in the all.mind input file is perturbed to the first and the third (page 38). These steps would be too large for an actual calculation (4.5° increments were used in *J. Am. Chem. Soc*. **114**, 7535 (1992)). Notice that there is no change in charge or other parameters along the pathway, thus the free energy change for the full rotational profile should be zero when a more meticulous calculation is carried out. This simulation is carried out in a box of 263 TIP4P water molecules, and 1000 steps of equilibration are performed.

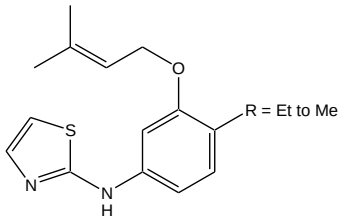
Test Jobs 21 and 22 – dAdcdT and 157d

These test jobs illustrate the construction of DNA and RNA polynucleotides with *pepz* and their subsequent energy minimizations.



Test Job 28 – FEP-RT

This illustrates the current, typical setup for performing an FEP calculation to compute a difference in free energies of binding. Two FEP calculations are run, one for the complex and one for the inhibitor alone in water. In both cases a 25-Å water cap is used. The specific example is for HIV-RT with inhibitors that have been synthesized in the Jorgensen lab.



Test Job 29 – polarize

This illustrates calculations including residue-residue polarization. The example is the complex of phenylalanine dipeptide with tetramethylammonium ion: (a) Fletcher-Powell optimization, (b) standard MC in water, and (c) MC/FEP in water ($Me_4N^+ \rightarrow Me_4C$).

Test Job 30 – FEP-ScanCl

This illustrates FEP calculations to scan for the optimal positioning of chlorines in a ligand for binding to a protein.

Test Jobs 31-33 - Overlap Sampling
Overlap sampling provides an alternative to standard double-wide sampling FEP calculations. See these test jobs and the *OverlapSampling* preprint in *MCPROman*.

Test Job 34 - JAWS
JAWS is used to optimize the placement of water molecules in a protein-ligand complex. See the testjob *neuraminidase* for detailed instructions on how to setup, execute and analyze a JAWS calculation. Utilities to facilitate analysis are in the folder *scripts*. The testjob *p38sos* illustrates how one would start an MC/FEP calculation using a JAWS derived solvent distribution.

Test Job 35 - HalogenBond
This illustrates the treatment of halogen bonding with the OPLS-AAx force field. Z-matrices are provided for halobenzenes and their complexes with acetone. A publication is also included in the MCPROman directory.

Test Job 36 – OPT-RT
This illustrates the CG-optimization of a ligand/HIV-1 RT –complex keeping large portions of the protein fixed.

19 Output

The output for Monte Carlo simulations in the "ot file" is in two main sections. Before the Monte Carlo calculation begins or resumes, the Z-matrix is listed along with the potential function and torsional parameters. The net charge of each residue in the reference solute is also reported; residues should usually have integral charges. The total energy and its components are recomputed from the coordinates in the in file or computed for the first time if ICALC = 1 or 2.

If it is a continuation run (ICALC = 0), then OLD E should equal NEW E at the beginning of the new run within the limits of precision of the computer (at least 7 figures). However, in MCPRO, the intermolecular residue–residue, solvent–residue, and intrasolute non-bonded pairs *lists are only computed at the very beginning of a run*. If they change with the start of a new run, EOLD may not equal ENEW on restarting. The lists are constructed initially larger than the cutoff radii (SCUT with a buffer of 2 Å), so the difference between OLD E and NEW E should still be small. (The program prevents the rejection of moves solely from a change in the lists.)

Many other parameters for the simulation are listed along with the energies. The values of RC0, RC1 and RC2 are then noted, followed by the coordinates for the reference solute and, if a perturbation calculation is being performed, for the two perturbed solutes—note that each of these two solutes may have multiple molecular fragments (sub-solutes). The Monte Carlo calculation then begins. When it finishes, the acceptance rate information for volume moves is printed. If a *ca.* 40% acceptance rate is not obtained, VDEL or RDELS/ADELS should be adjusted in the parameter file, or the dihedral angle variations in the Z-matrix may be modified. The following output is provided for Monte Carlo runs:

1. plots showing the history for up to 8 variable dihedral angles in the solutes when IPRNT > 2,

2. the final total energy and its components along with the parameters for the simulation as above,

3. the final coordinates (optional) and solvent-accessible surface area and volume for the solutes,

4. the averages for the thermodynamic properties including the two free energies which are repeated in fuller form below (Note: ESX includes the CAPPOT energy if applicable),

5. the solute–solvent and solute–solute Coulombic and Lennard–Jones components of the energy (the cap potential is NOT included here),

6. the solvent–solvent atom–atom radial distribution functions and their integrals (coordination numbers) that have been requested in the parameter file,

7. the solvent–solvent and solute–solvent total energy and energy pair distribution functions,

8. the distribution functions for the variable dihedral angles,

9. the record of attempted and accepted moves for each residue, solvent and solute molecule, and

10. the full report on the computed thermodynamic results including the averages for each run, the total averages and the standard deviations (1σ) calculated from the fluctuations in the averages for each run *(i.e.,* the standard error of the mean). This output is provided when more than one run has been completed.

Note: results are included on the ΔH and ΔS for the two ΔG's that are computed in perturbation calculations. These are computed by an umbrella sampling procedure. Unfortunately, the statistical noise is so large that the computed ΔH's and ΔS's are usually too imprecise to be useful.

It should be noted that the average of a property over the entire simulation may or may not be equal to the average of the averages for each run (block). This equality holds for properties given by linear functions such as the total energy, energy components, and volume. However, it does not hold for properties given by non-linear expressions including the fluctuation properties (heat capacity, coefficient of thermal expansion, and compressibility) and the free energy changes from the Zwanzig equation. In the latter case,

$$\Delta G_{avg} = -kT \, ln \, \{1/NRUN * [\sum_{i} \exp(-\Delta G_i / kT)]\}$$

where the summation is over i = 1 to NRUN, the number of blocks.

When the command file requests multiple runs on one job submission, our practice is to name the resulting multiple ot files xxxota, xxxotb, etc. where xxx is the 3 letter identifier for the project. **Note:** Abbreviated output is provided for solute optimizations (ICALC = 2). Definitions for some of the output quantities are given below. All energies are in kcal/mol, all distances in Å, and all angles in degrees.

19.1 Some Variable Definitions

NMOL = The number of solvent molecules.
EDGE = The dimensions of the simulation cell.
NCENT = The centers defined for solute 1 and 2 (NCENT1, NCENT2).
RCUT = The solvent–solvent cutoff distance.
SCUT = The solute–solvent cutoff distance.
NACCPT = The total number of Monte Carlo steps (configurations).
NRJECT = The number of attempted steps that were rejected (*ca.* 60% NACCPT).
MXCON = The number of steps requested for the current run.
NCON = The current step number for this run.
T = The temperature in °C.
P = The external pressure in atm.
RDEL = The range for solvent translations.
RDELS1 = The range for translations of solute 1.
RDELS2 = The range for translations of solute 2.
ADEL = The range for solvent rotations.
ADELS1 = The range for rotations of solute 1.
ADELS2 = The range for rotations of solute 2.
OLD E = The total energy for the last accepted configuration.
NEW E = The total energy for the last attempted configuration.
VOLD = The total volume in $Å^3$ for the last accepted configuration.
VNEW = The total volume for the last attempted configuration.
NSCHG = Frequency of attempted solute 1, 2 moves, in configurations (page 19).
NVCHG = Frequency of attempted volume changes.
DIELEC = The dielectric constant.
ESOLD = The old (last accepted) total solute–solvent energy.
ESOL1, 2 = The same for the two perturbed solutes.
ESONE = The new (last attempted) total solute–solvent energy.
ESON1, 2 = The same for the two perturbed solutes.
EXXOLD = The old solute–solute energy.
EXXOL1, 2 = The same for the perturbed solutes.
EXXNEW = The new solute–solute energy.
EXXNE1, 2 = The same for the two perturbed solutes.
EDIHOL = The old torsional energy for the solutes.
EDIOL1, 2 = The same for the perturbed solutes.
EDIHNE = The new torsional energy for the solutes.
EDINE1, 2 = The same for the two perturbed solutes.
ENBOL = The old intrasolute non-bonded interaction energy.
ENBOL1, 2 = The same for the perturbed solutes.
ENBNE = The new intrasolute non-bonded interaction energy.
ENBNE1, 2 = The same for the two perturbed solutes.
EBCOLD = The old energy for the distance constraints.
EBCOL1, 2 = The same for the perturbed solutes.
EBCNEW = The new energy for the distance constraints.
EBCNE1, 2 = The same for the two perturbed solutes.
CUTOFF E = Non-aqueous solvent–solvent energy correction for Lennard–Jones interactions neglected beyond the cutoff RCUT.

20 Contents of the Distribution Files

File Name	Description
README	First file to read.
MCPRO or	Linux executable.
MCPRO.exe	Windows executable.
ViewerLite42.exe	A freeware version of Windows WebLabViewer - install by double-clicking.

MCPROman/ Directory containing the user's manual and reference publications.
The user's manual is in *MCPROman.pdf*

molecules/
small/ Z-matrices for small organic molecules.
drugs/ Z-matrices for drugs.
peptide/ Z-matrices for peptides.
notes/
parfile.format Template for MCPRO parameter file header.
Zmatrix.format Template for MCPRO Z-matrix file format.
Other files with miscellaneous notes.

miscexec/ All executable programs except for MCPRO.
solbox/
watbox File of standard water boxes.
org1box File 1 of standard non-aqueous solvent boxes.
org2box File 2 of standard non-aqueous solvent boxes.
wcapin.18 Equilibrated 18 Å radius water cluster.
wcapin.20 Equilibrated 20 Å water cluster.
wcapin.22 Equilibrated 22 Å water cluster.
wcapin.25 Equilibrated 25 Å water cluster.
probe2A.in Equilibrated 15 Å cluster of 2-Å probe spheres.

AA/
oplsaa.par The latest version of the OPLS all-atom parameters.
oplsaa.sb All-atom stretching and bending parameters.
oplsaa.db All-atom database file for *pepz* program. See Appendix 1.
dihedrals.aa All-atom dihedrals file for *pepz* program. See Appendix 1.

UA/
oplsua.par The latest version of the OPLS united-atom parameters.
oplsua.sb United-atom stretching and bending parameters.
oplsua.db United-atom database file for *pepz* program. See Appendix 1.
dihedrals.ua United-atom dihedrals file for *pepz* program. See Appendix 1.

source/ Normally not included.
mcpro.f FORTRAN source code for MCPRO.
miniconrot.c C code for conrotation.

gbsam.f	GB/SA code.
propCMP.f	Properties prediction code for CM1P charges.
propAA.f	Propertis prediction code for OPLS-AA charges.
Makefile	Contains compiling macros.

pepz/	Normally not included.
Makefile	Contains compiling macros for *pepz*. See Makefile for listing of FORTRAN subroutines and included files for *pepz.*

pepz/FreeRead/

Makefile	Contains compiling macros for the freeread library, libFree.a, needed for *pepz*. See Makefile for listing of FORTRAN subroutines needed for library.

pepz/Utilities/

fixpdb.f	Makes PDB files which have non-contiguous residue or atom numbering schemes contiguous so *pepz* will be able to use them.
pdb2newzm.f	Short program that simply writes a new Z-matrix given a plt (PDB) file and an old Z-matrix. This may be a useful first-step for building a new ligand or complex Z-matrix from the final simulated conformation of an analogous system, if the topology of the system does not change. Atom names and residue numbers from the old Z-matrix are matched to PDB atom (residues) information, and all bonds, angles, and dihedrals of the old Z-matrix are replaced by those calculated from the PDB coordinates. All atom type and variation information is copied directly from the old Z-matrix to the new Z-matrix.

21 The pepz Program, A Z-matrix Builder for Biomolecules

21.1 Introduction

The *pepz* program was written to facilitate the building of MCPRO Z-matrices for peptides, proteins, protein-ligand complexes, and polynucleotides. The current version of *pepz* works as a batch utility; that is to say, it will run from the command line, the instructions to the program have to be previously stored in an input file, and it will create one or more output files. Efforts are underway to integrate it with a graphical user interface. Briefly, the program can be used by issuing from the UNIX prompt the command

pepz –i *name_of_input_file* –o *name_of_output_file* [-v]

where the input file could be as minimal as:

```
$ title Capped undecaalanine peptide
$ read database oplsaa.db
$ read dihedrals dihedrals.aa
$ read parameter oplsaa.par
$ sequence ace ala ala ala ala ala ala ala ala ala ala ame
$ write zmatrix ala11.zmt
```

The preceding file would write in a file called ala11.zmt a complete Z-matrix for an all-atom undecaalanine peptide, capped with acetyl and N-methyl amide groups in an extended conformation where all the bond lengths, angles, and torsions are designated to be variable. The OPLS united atom and all-atom non-bonded parameters and torsional Fourier coefficients, which were designed to reproduce the *ab initio* 6-31G* rotational profiles, are used. Examples of the use of *pepz* for larger systems are included in the Test Jobs.

21.2 Commands

The input to *pepz* can contain three different type of lines: comment lines that start with a pound sign (#), command lines that start with a dollar sign ($), and data lines. The commands in the following table are recognized by *pepz*:

Command	Qualifier	Qualifier	Qualifier	Arguments
title				string
read	database			filename
read	parameter			filename
read	dihedral			filename
read	boss			filename
set	parameter	type	all-atom	list of residue numbers
set	parameter	type	united-atom	list of residue numbers
set	conrot			[*termini*]
set	override	domains		list of residue numbers
sequence				blank-separated residue (3-letter code) list
read	sequence			filename
read	pdb			filename
ssbond				pairs of residue numbers

center				
set	fixed		bonds	list of residue numbers
set	fixed		angles	list of residue numbers
set	fixed		improper	list of residue numbers
set	fixed		unsaturated	list of residue numbers
set	fixed		backbone	list of residue numbers
set	fixed		all	list of residue numbers
set	fixed		none	list of residue numbers
set	variable		bonds	list of residue numbers
set	variable		angles	list of residue numbers
set	variable		improper	list of residue numbers
set	variable		unsaturated	list of residue numbers
set	variable		backbone	list of residue numbers
set	variable		all	list of residue numbers
set	variable		none	list of residue numbers
set	preferred			list of residue numbers
write	pdb			filename
write	mind			filename
write	zmatrix			Filename

A detailed explanation of each command is given below, but a few general comments regarding syntax are needed. First, each line in the input file is parsed using the freeread library and is executed in order before the next line is parsed, although long arguments (*i.e.*, sequence lists) can be continued in subsequent lines and are gathered by the program before execution. *Long lists of residue numbers may be shortened by repeating the command on the next line and continuing the residue list*. There is also limited support for an instruction queue, so some of the commands that can not be executed at the exact moment of their parsing are held in a queue for processing when enough information about the protein has been gathered or generated by the program. *In general, commands should appear in the order listed above*; the reading of the database file, parameter file, dihedrals file, and additional Z-matrix (if any) should occur before any other commands. As all the commands and data are read in free format, they can be in any column in the input line. They are also case independent, and the commands and qualifiers can be abbreviated down to a minimum of three letters.

A list of residue numbers can be given in several ways; a series of disconnected numbers separated by either blanks or commas, a range indicated by the initial and final residues separated by either a colon or a dash (*e.g.*, `2:12` or `2-12`), and finally the keyword "all" or the wildcard "*" can be used to indicate all the residues in the sequence.

- title *string* — The string following the keyword will be used as the header in the Z-matrix.

- read database *filename*

  The program *pepz* requires a database of templates for each residue which are combined to form the final Z-matrix. The database files, oplsua.db and oplsaa.db, that implement the united-atom and all-atom OPLS force fields are included in this distribution. This line is required. Multiple database files can be read by using additional "read database" lines.

- read parameter *filename*

  The parameter file to be used in the MCPRO simulation. It is needed by *pepz* only to establish a translation table between the atom types and their two-letter codes employed in the assignment of dihedral parameters below. This line is required.

- read dihedral *filename*

  This file provides a way to assign the dihedral angle type (from the parameter file) based on either the atom types or the two letter codes to all the torsional angles in the Z-matrix. United-atom and all-atom versions are included in the distribution. The file is required only if dihedral angles are varied. Multiple dihedrals files can be read by using additional "read dihedral" lines.

- read boss *filename*

  pepz can also read a single Z-matrix from either the BOSS or MCPRO Monte Carlo programs. This is particularly useful when the substrate or inhibitor of an enzyme is a small organic molecule. A PDB file is usually read for the correct docking of the ligand in the protein structure; multiple solutes in the Z-matrix will be recognized if they appear in the PDB file. Only one Z-matrix can be read at a time, and each solute should be a single residue. Atoms will be matched based on atom name and residue number, and thus each atom name must be unique. Inhibitors can be divided into multiple residues manually after *pepz* is run.

  Any dummy atoms in this Z-matrix will be retained in the *pepz* Z-matrix and should not be given residue number 0—this results in confusion with the initial dummy atoms in the *pepz*-built Z-matrix.

  Variable and additional bond, angle, and dihedral information will be renumbered and incorporated correctly in the final Z-matrix but the order in which they appear in each section may differ from that given in the input Z-matrix.

- set parameter type [united|all] *argument*

This new command is needed to differentiate between united-atom and all-atom residue templates if oplsaa.db and oplsua.db are concatenated into one opls.db file or if both database files are read. The program will look for the parameter type in the residue template header, so this command must come *before* the sequence information is read and new residues are added from templates to the Z-matrix.

- set conrot *termini*

  This command sets the necessary flags to allow the movement of backbone dihedrals *via* the concerted rotation algorithm. If the optional *termini* keyword is present, the backbone atoms of the 3 (1) terminal residues for polypeptides (polynucleotides) on both chain ends are NOT sampled.

- set override domains *argument*

  This new command is required to override the default use of domain definitions in building the Z-matrix. Certain degrees of freedom (*i.e.*, the bonds, angles, and dihedrals in aromatic rings) are not sampled when domains are used, but for fully-flexible Z-matrices all degrees of freedom must be variable. As this command influences many degrees of freedom, it must be given in the input file *before* the sequence information is read, as residues and their associated degrees of freedom are sequentially added to the Z-matrix.

- sequence *argument*

  A list of residues as a sequence of standard three letter abbreviations. The "TER" keyword can be used to separate chains that are not covalently linked. This line is not required since it can also be read from a sequence file or directly from the pdb file (see below); however, it is the only way of adding residues that are not present in the pdb file or for which Cartesian coordinates are not available. (Examples would be additional ACE or AME sequence capping groups.)

- read sequence *filename*

  This is not required, but useful if a very long sequence is used. The file should contain ONLY the sequence.

- read pdb *filename*

  This line is only required if a coordinate set from another source will be used. The values of the internal coordinates of all the atoms present in this file are calculated and, whenever possible, substituted for the idealized values present in the original Z-matrix. Atom and residue names in the PDB file must match those in the database file or BOSS/MCPRO Z-matrix to be included. The order of the atoms in the PDB file does not have to match that of any included BOSS/MCPRO Z-matrix; however, the atom and residue numbering must be contiguous. The *fixpdb* utility program included may be useful for converting a non-contiguous PDB file for use with *pepz*.

- ssbond *argument*

  This keyword creates disulfide bridges among the residue pairs specified in the argument.

- center

  In the normal process of building a Z-matrix, three non-interacting (dummy type –1) atoms are placed at the origin of the Cartesian axis and their connections to the real atoms are calculated in such a way as to preserve the absolute positions in space defined in the pdb file. This keyword directs the program to move the dummies to the geometric center of the molecule(s) instead. The relative positions in space of all the real atoms are preserved by this transformation.

- set [fixed|variable] [bonds|angles|improper|unsaturated|backbone|all] *argument*

  These commands allow the selection of degrees of freedom to be sampled during the simulation. They can be specified in any order to produce the desired combination of variables. If none of these directives is specified, the default is to allow all degrees of freedom to be sampled. **Note:** the implementation of the united-atom force field in *pepz* was designed for use with "set fixed unsaturated all".

- set preferred *argument*

  For use with the united atom parameter set only! All bonds are fixed, all angles are fixed except for those around the backbone amide nitrogen and the sidechain nitrogen of lysine, and all the dihedrals are variable. We have found this mode to be a good compromise between freedom of motion and CPU time while preserving adequate sampling.

- write pdb *filename* — This keyword directs the program to write a pdb file containing not only the real atoms but also the dummies. It is useful if one wants to examine or modify some specific positions or variables before the actual simulation begins. The modified file could then be read by the "read pdb" command to generate a corresponding Z-matrix file.

- write mind *filename* — This command will cause writing the Cartesian coordinates to a file in the format needed by our graphics programs MindTool and *yummie*. This format contains the atomic number, atom name, Cartesian coordinates, and the residue name and number so it is very easy to convert into the format required by other display or analysis programs.

- write zmatrix *filename* — This line allows the specification of a name for the Z-matrix produced by *pepz*. It is not required, but no file will be written if it is absent.

21.3 Database and Dihedrals Files

In addition to the input file that contains the commands, *pepz* requires other files as described above. The **database files** (oplsua.db and oplsaa.db in the standard distribution) contain a series of template Z-matrices; united-atom histidine is shown below as an example. These files also contain the assignments of atom type which will be used to look up the non-bonded parameters, and the default (idealized) geometry for all the atoms. **Note**: *the atom types must match those used in the parameter file or incorrect parameters will be used in the simulation. Be sure to use oplsua.par with oplsua.db and dihedrals.ua, for example.*

Briefly, the database file is organized in three separate sections for residues at the N-terminus, in the middle of the chain, and at the C-terminus. Each residue is initiated by a header line that contains the three letter code, the number of atoms in the residue, the number of additional bonds (typically used for ring closures), the number of additional dihedrals (a new feature, necessary for maintaining planarity in fully-flexible rings), the number of excluded domains, the number of excluded regions, the parameter set used (OPLS in the current database files), the type of parameters (UNI for united-atom, ALL for all-atom), and a chain position indicator (N, M, or C). This line is read with a FORTRAN format of (`a3,5i4,1x,a10,1x,a3,1x,a1`).

The following lines, one for each atom, contain the specification for each atom as follows: atom name, atom type, atom to which it forms a bond (atom name and relative residue number), the default length for the bond, atom to which it forms an angle (atom name and relative residue number) followed by the default value for the angle, atom to which it forms a torsion (atom name and relative residue number) followed by the default value of the torsion, an optional alias (an alternative atom name), and finally a single character indicating a topological type. This line is read with a (`2x,a3,2x,i3,3(x,a3,x,i3,2x,f10.4),x,a3,a1`) format.

Please note: The templates follow PDB guidelines for assignment of atom names and stereochemistry, however, experimental structures do not generally have hydrogen atoms and do not always follow "the rules". With some recent examples in our laboratory, non-standard assignment in crystal structures of "equivalent" γ-carbon atoms in **valine** residues and δ-carbons in **leucine** residues has resulted in collinear positioning of HB or HG along the CB-CG2 or CG-CD2 bonds in all-atom Z-matrices. Please check the output Z-matrix carefully. This is addressed further in Test Job CDK2.

If there are any additional bonds they are read next, one line for each bond. The additional bond must be specified as name of atom i, its relative residue number, name of atom j, and its relative residue number. These lines are read with a (`a3,i5,x,a3,i5`) format. There is no need to have any lines in this section if the residue does not contain any additional bonds.

If there are any additional dihedrals (atoms i-j-k-l), they are read next, one line for each dihedral. Currently, these will only be found in the all-atom database file. (The united atom template residues assume rigid aromatic rings, etc. and do not have high-barrier torsional parameters defined to maintain planarity in flexible systems.) The additional dihedral must be specified as name of atom i, its relative residue number, name of atom j, its relative residue number, the name of atom k, its relative residue number, and the name of atom l and its relative residue number. These lines are read with a `(a3,i5,3(x,a3,i5))` format.

The following lines define the excluded domains. An excluded domain is a pair of atom sets (the first set from atom i to atom j and the second from atom k to atom l) whose interatomic non-bonded energies are not evaluated. It is normally used for portions of the molecule whose relative positions do not change in the simulation and thus their interatomic energy remains constant. A line for each domain containing the name of atom i, relative residue number, name of atom j, relative residue number, name of atom k, relative residue number, name of atom l, relative residue number is read with a (`a3,i5,3(x,a3,i5)`) format.

The very last section of a residue is the specification of excluded regions—a special case of excluded domains in which all the atoms are contiguous in the Z-matrix, *e.g*., aromatic rings which are moved as rigid bodies. These are specified as a line for each region containing the name of atom i, relative residue number, name of atom j, and relative residue number. These lines are also read with a (`a3,i5,x,a3,i5`) format.

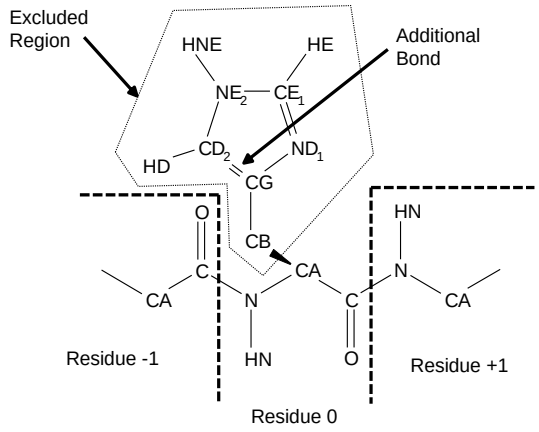
This information on excluded domains and regions for residues is used to construct the "Domain Definition" portion of a MCPRO Z-matrix; the "Excluded Atoms List" section of a Z-matrix is constructed within *pepz* when *set fixed backbone* is specified. As in the case for additional bonds, there is no need to add blank lines if the residue does not contain any excluded domains or regions. In the database files provided in the distribution there is a blank line at the end of every residue for clarity, but it is not required.

The portion of the file corresponding to a united-atom histidine in the middle of the chain is given below as an illustration:

```
HIS  14   1   0   0   1 OPLS       UNI M
  N      3 C    -1      1.335  CA   -1      116.60 N    -1       180.0    M
  CA     6 N     0      1.449  C    -1      121.90 CA   -1       180.0    M
  C      1 CA    0      1.522  N     0      111.10 C    -1       180.0    M
  HN     4 N     0      1.010  C    -1      119.80 CA    0       180.0    L
```

```
  CB   304 CA    0      1.555  N     0      111.10 C     0      -120.0    S
  CG   381 CB    0      1.517  CA    0      117.30 N     0       180.0    S
  ND1   42 CG    0      1.390  CB    0      122.00 CA    0       180.0    S
  CE1  380 ND1   0      1.320  CG    0      108.00 CB    0       180.0    S
  NE2   40 CE1   0      1.310  ND1   0      109.00 CG    0         0.0    S
  HE   231 CE1   0      1.090  ND1   0      120.00 NE2   0       180.0    E
  CD2  382 NE2   0      1.360  CE1   0      110.00 ND1   0         0.0    S
  HNE   41 NE2   0      1.010  CE1   0      125.00 CD2   0       180.0    E
  HD   231 CD2   0      1.090  NE2   0      120.00 CG    0       180.0    E
  O      2 C     0      1.229  CA    0      120.50 N    +1       180.0    L
CD2    0 CG     0
CB     0 HD     0 CB     0 HD     0
```

A representation of the structure along with the nomenclature used is shown below to illustrate some details of the Z-matrix. First, the division of the atoms in topological classes (the last item in the entry for each atom) in the residue can be seen more easily. Basically, the backbone atoms sequentially connected (N, CA, and C) are in the main chain class **M;** the heavy atoms traditionally classified as side-chains (CB, CG, ND1, CE1, NE2, and CD2) are in the sidechain class **S**; atoms directly connected to the backbone which are not side-chains are in the lateral class **L**; and atoms that branch from the sidechain and are defined by improper dihedral angles are in the external class **E**. This classification is used by *pepz* for the selection of variable degrees of freedom. The atoms enclosed in the polygon form the excluded region (all the atoms in the imidazole ring and those directly connected to it).



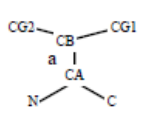
Another important detail of the way in which the template Z-matrices are constructed is the use of improper dihedrals to build some of the atoms, *e.g.*, CB(0) is defined from CB(0)–CA(0)–N(0)–C(0) instead of –C(–1). In this case, the improper dihedral is used to maintain the stereochemistry around the united-atom CA without the need to define additional force constants, as is the case in MD. The energy from the real dihedral interactions around the –CA–N– and –CA–C– bonds is evaluated through the use of "additional dihedrals" later as *pepz* builds the Z-matrix. Improper dihedrals are also consistently used for defining rigid-body rotation about single bonds (see discussion page 31).

Lastly, the **dihedrals files** (dihedrals.ua and dihedrals.aa in the standard distribution) provide a way to assign a specific torsional type from the parameter file to a given set of atoms. It contains one line for each assignment that specifies the MCPRO atom types of atoms i, j, k, and l, optionally followed by their two letter (AMBER) codes, the torsional type from the parameter file, and an optional integer 1,4-scaling factor (SCL14C = SCL14L). The format used to read this file is `(4(i4,1x),2x,4(a2,1x),2x,i4,2x,i4)`, so the vertical separators and comments at the end of the line are not read by *pepz*. Additionally, lines starting with the pound sign (#) are treated as

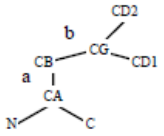
comments. A small section of the distribution file corresponding to the backbone of most united-atom amino acids is given here as an example:

```
#                    |              |       |      | for united atom OPLS:
   1-   3-   6-   1 | C -N -CH-C   |   47 |    2 | phi
   1-   3-   6-   7 | C -N -CH-C3  |   48 |    2 | phi-C
   3-   6-   1-   3 | N -CH-C -N   |   49 |    2 | psi
   3-   1-   6-   7 | N -C -CH-C3  |   50 |    2 | psi-C
   6-   1-   3-   6 | CH-C -N -CH  |   75 |    2 | CA-C -N -CA     Omega, all AA
```

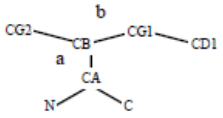
The program sets dihedral angle variations (PDELS) in the Z-matrix to 1 and 4° for unsaturated and saturated main chain angles, respectively, 2° for impropers, and 7 or 15° for sidechain torsions. We have found these ranges to perform well, but they could be modified manually to increase acceptance ratios in a MC simulation. Sample PDELS for united-atom amino acids are shown below, along with excluded regions defined in the database file. Additional residues in oplsua.db are GLY, α-amino isobutyric acid (AIB), cystine (CYX), hydroxy- lysine and proline (HYL, HYP), phosphotyrosine (PTY), and the acetyl (ACE), N-methyl (AME), and amino (NH_2) "capping" groups. The all-atom database includes all standard residues and capping groups, along with CYX (half-cystine), HID (HIS-δ), HIP (protonated HIS), and the neutral forms of acidic and basic residues, ASH, GLH, LYN, and ARN.



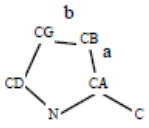
VAL a = 15



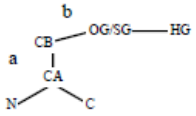
LEU a = 10
b = 15



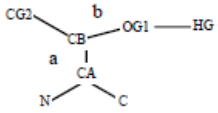
ILE a = 10
b = 15



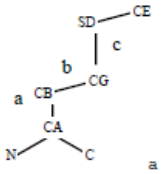
PRO a = 2
b = 2



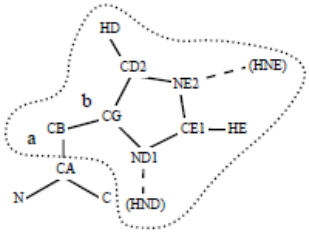
SER/CYS a = 10
b = 15



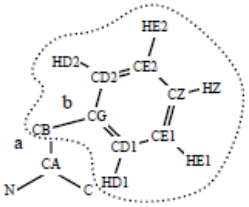
THR a = 10
b = 15



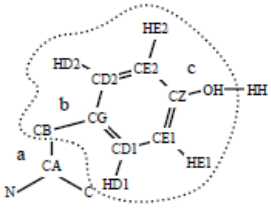
MET a = 5
b = 10
c = 15



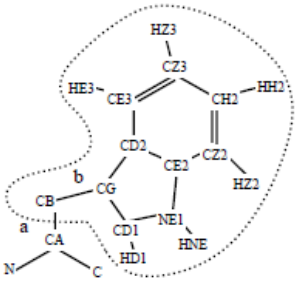
HIS (HNE), HID(HND), HIP (HND,HNE) a = 5
b = 10



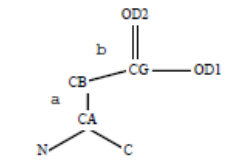
PHE a = 5
b = 10



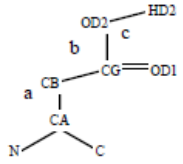
TYR a = 5
b = 10
c = 15



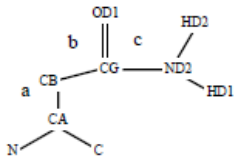
TRP a = 5
b = 10



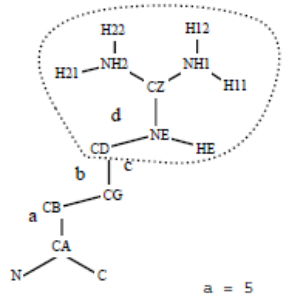
ASP a = 10
b = 15



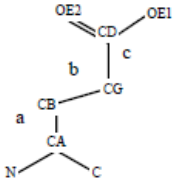
ASH a = 10
b = 15
c = 15



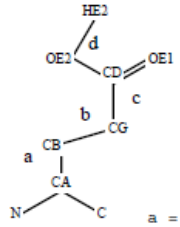
ASN a = 10
b = 10
c = 15



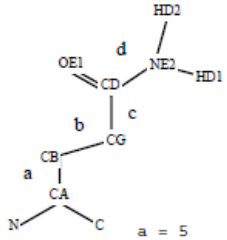
ARG a = 5
b = 5
c = 10
d = 10



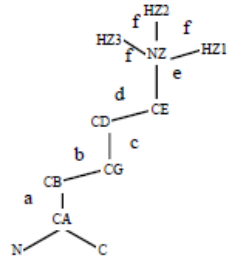
GLU a = 5
b = 10
c = 15



GLH a = 5
b = 10
c = 15
d = 15



GLN a = 5
b = 10
c = 10
d = 15



LYS (imp.) a = 5
b = 5
c = 10
d = 10
e = 15
f = 5

22 Appendix 1 - User's guide to *chop*

The chop program is a utility designed to facilitate many of the operations needed in order to run MCPRO calculations of proteins. Most of them involve laborious, tedious, and error prone manual editing of large pdb files. Chief among them are carving an adequate number of residues from the entire protein, selecting an origin for the water cap, setting charge and tautomeric states of some residues, and the renumbering of atoms and residues. The chop program is designed to be run from within UCSF midas and, taking advantage of midas graphics and selection utilities, perform most of those operations within a single graphical environment.

Although the midas program does lack a real scriptable language, it does provide a mechanism to interact with external program called the "delegate" utility. This mechanism is completely described in their manual but briefly, the delegate utility sets a name or prefix (chop in this case) that when used at the beginning of a line will cause the remainder of the line to be sent as input to the delegate program. After the line is sent midas reads the output generated by the delegate program as a midas command. One thing that it does not provide, unfortunately is the direct communication of data to and from the program so a lot of care must be exercised in order to ensure the synchronization of the actual data residing in the delegate program and the graphical display as shown by midas.

As an illustration of its usage, a typical session is given below for the preparation of the pdb file and pepz inputs for an active site model based in the crystal structure of HIV protease complexed to one of the SK&B inhibitors:

	Command typed	**Result**
1	midas	*starts midas*
2	delegate start chop chop –i 1aaq.pdb	*starts the chop program and defines "chop" as the prefix for commands to be sent to delegate. It opens the pdb file.*
3	chop add center :psi	*adds center of ligand (:C01@C01)*
4	chop set cap origin :c01	*makes this center the origin of water cap*
5	chop set cut origin ligand	*define cut from all atoms in ligand*
6	chop set cut size 15	*cuts residues within 15Å*
7	chop fix chains	*completes chains*
8	chop cap all	*adds Ace and Ame neutral caps*
9	chop set variable origin ligand	*define variables from all atoms in ligand*
10	chop set variable size 10	*makes residues beyond 10Å fixed*
11	chop set targetq -1	*the charge in the ligand is +1, but neither midas nor chop know that!*
12	chop fix charges	*neutralizes enough residues to reach the target charge of -1*
13	chop write pdb 1aaq.chop.pdb	*writes the new pdb file*
14	chop write pepz all 1aaq.chop.all	*writes pepz input for minimization*
15	chop write pepz variable 1aaq.chop.var	*writes pepz input for simulation*
16	chop write translation 1aaq.chop.tt	*writes translation table file*
17	quit	*stops both midas and chop*

The chop program works by defining and keeping track of sets of residues and atoms through various types of operations. Each time a set is defined or modified one or more commands are sent to

midas to produce a display that will reflect the results. There are three basic residue sets, the "ligand" set contains the residues that will being part of the ligand will be allowed full flexibility in the MC simulations. The "cut" set contains the residues to be included in the model (*i. e.* you cut-in the residues you want to keep), and the "variable" set contains the residues that will be allowed to move during the simulations. Each of these sets are defined initially by selecting an atom set as origin and a distance from this atom set, the "cut origin" and "variable origin" sets and the "cut size" and "variable size" distances, respectively. Further refinement can be done on each set by adding or deleting residues. The additional atom set "ligand" is also used for display and analysis purposes.

Following the illustration session given above, when the command in line 2 "delegate start chop chop –i 1aaq.pdb" is typed in midas the following events occur in order. First midas defines the word chop as a prefix to indicate that whenever a command starts with that word the remainder of the line should be sent to the chop program. Any word that does not conflict with midas commands or aliases can be used instead of chop. It then starts the program with the command line arguments "–i 1aaq.pdb". Chop opens that file, reads it, initializes some arrays and parameters, and then sends midas the commands needed to open the file as model 0, display the C^{α}-trace, the complete ligand if it could recognize it based in HETATM records, color it, and label the ends of the chains.

When commands 3-16 are typed in midas, the chop word is recognized as the prefix for the delegate, and the remainder of the line is sent to the chop program. In command 3, "chop add center :psi" chop calculates the geometric center of the residue named PSI and add a dummy atom at that position, writes a file containing its coordinates and instructs midas to read it as model 1. Note that it is not necessary to type the residue name. Alternatively, one could just type "chop add center" and pick any atom in that residue (colored green), remove the atom specification (the portion after the @) and hit <ENTER> This capability extends to all the commands that require an atom or residue specification as arguments. In all cases, each added center is defined as a separate residue named Cxx that contains a single atom, also named Cxx, where xx is a two digit sequential number. Command 4, "chop set cap origin :c01", creates another dummy atom labeled CAP at the position of the center just defined. The difference between this CAP and the centers is that the cap will be written to the final pdb and pepz files, while the centers will not. In addition, by defining a cap size (*i.e* . "chop set cap size 22") a sphere of that radius will be shown in the screen as a visual approximation of the size and location of the water cap to be used in the simulation.

Command 5 and 6, "chop set cut origin ligand" and "chop set cut size 15", define the reference atom set cut-origin to contain all the atoms in the previously defined ligand set and select all the residues with at least one atom within 15 Å of the ligand to be included in the cut. The display is then refreshed and the residues to be excluded are drawn in colors darker than the ones included. The text line will give a small informational message including the number of residues included in the cut and the current charge. The residues at the termini of the newly created chains are labeled, but additional information can be obtained by the commands "chop status charge" or "chop status chains". Issuing additional commands "chop set cut size XX" will change the sets and the display accordingly. Note that although the chop delegate always keeps track of the correct count and sequence number of residues, the original residue numbers and chain designators as read from the pdb file are maintained during a chop session for ease of operation. It would be rather taxing to the user if the residue names and numbers were changing continuously, but the correct, sequential residue numbers are written to the pdb file at the end of the session. Disulfide bridges, if present and specified in the input pdb file, are also automatically handled by the chop delegate.

The typical concern at this point is to minimize the number of disconnected chains. That can be done manually by using the commands "chop add cut Residue_Spec" or "chop delete cut Residue_Spec", but an automatic procedure in chop is invoked by command 7, "chop fix chains. Briefly, the algorithm involves looping over all the chains in the current representation and eliminating all the gaps of 3 residues or less by adding those residues to the cut set. A second loop then deletes from the set isolated chains of 2 residues or less as long as they are within 2.0 Å of the cut distance. All these parameters (mingap, minchain, and threshold) can be modified by the user. Command 8, "chop cap all", then causes the substitution of **ALL** terminal residues by the appropriate acetyl on N-methyl amide groups. Note that the labels are refreshed, but the atom name remains as CA. This is one of the side effects of the lack of direct communication of data between midas and the delegate, but the atom names of the termini are correctly written to the pdb file by the delegate. The cut set should not be modified after this point because chains cannot be lengthened or shortened after being capped. If further modification is needed use the "chop uncap all" command before any other commands.

Once an appropriate system size has been achieved and all chains have been terminated, commands 9 and 10, "chop set variable origin ligand" and "chop set variable size 10", are used to define the set of variable residues in the same manner than commands 5 and 6 were used to define the set of residues included in the cut. This set can be modified by adding or deleting residues, but there are no commands implemented to automatically modify it.

The next issue to address is the total charge in the system, which must typically be neutral for most types of simulations. This can be achieved manually by the commands "chop neutralize Residue_Spec" or "chop charge Residue_Spec", but commands 11 and 12 use a built-in facility to do so easily. First, command 11 "chop set targetq -1", is used to specify that the total charge to achieve is -1 (the default is obviously 0) because this particular ligand has a positive charge. Command 12, "chop fix charges", then attempts to neutralize enough Asp, Glu, Lys, Arg, and Hip residues to achieve this goal. The mechanism of this procedure is rather conservative and tries not to modify the charge near the ligand. Briefly, the entire system is divided in four zones based in half of the difference between the cut size and the variable size, (15-10)/2= 2.5 Å in this example. The outermost of these zones includes all the residues that are between 12.5 Å and 15 Å halfway from the ligand. All residues in this zone are neutralized. The next layer inwards includes residues that are between 12.5 Å and 10 Å, the edge of the variable zone. If at all possible, the system is adjusted to reach the target charge by neutralizing as many residues as possible in this zone to achieve that goal. If no further adjustments are needed all variable residues are kept in their original charge state. In cases where it is needed, the minimal number of residues in the next layer (10 to 7.5 Å) will be neutralized in order to reach the desired charge. If the target charge cannot be achieved after this, a warning message is typed in the screen, but in no case will the charge state of the innermost residues (7.5 Å or less from the ligand in this case) be changed. The remaining charge adjustment can be done manually by the user by using the appropriate charge and neutralize commands of the chop delegate. It must be noted also that the automatic algorithm will not charge or neutralize residues that have been previously modified by the user in the course of the session.

This particular protein contains only two histidines and they are both quite far from the ligand, so there was no need to decide on their protonation or tautomeric states. If that is not the case, the "chop charge Residue_Spec" command can be used to change a His to a Hip, while "chop neutralize Residue_Spec" will do the reverse. If desired, the exact nature of the residue can be also changed by the commands "chop set {hid|hie|hip|his} Residue_Spec". An additional command, "chop set view his 1 (or 2 or …)" will instruct midas to display and label only those residues within 4.5 Å of the first (or second

or …) His residue in the current view. The normal display can then be restored by the command "chop set view default".

The remaining commands are used for output and to exit. Commands 14 and 15 will both produce two input files for pepz. The difference between them is that the input written by 14, "chop write pepz all 1aaq.chop.all" instructs pepz to write a z-matrix in which all degrees of freedom are variable. This z-matrix can be used for the initial conjugate gradient minimization. Command 15, "chop write pepz var 1aaq.chop.var", will produce an input file that includes the commands needed by pepz to write a z-matrix in which the ligand is fully flexible and only the sidechains of the residues selected as variable in the current chop session are allowed to move.

Although the chop program was designed to be run within midas, it can be also run from the command line or a shell script if so desired. The biggest differences are that the command line flag "-u" should be used to bypass the midas synchronization mechanism, the "delegate start" command is not needed, the chop prefix to all commands should not be used, and that graphical selection of atoms and residues is obviously not possible. There is no exit command, instead the program exits when it encounters the End-of-File condition if a file is used as input or the EOF character (<CTRL>D in Unix). The complete calling syntax is:

$MCPROdir/chop/chop [-u|-q|-c] [-v] [-n] [-r record_file] [-i pdb_file]

-u: the chop program will interact directly with the user, bypassing the midas synchronization mechanism
-q: the program will issue a series of questions (a query) to define all needed information.
-c: checks the input PDB file and exits.
-v: **v**erbose, it will produce copious debugging output
-n: does **n**ot write protein hydrogen atoms if they are present in the input pdb file. This will allow pepz to place them in the default positions.
-r **r**ecord_file: writes a file containing all the instructions received and all responses by midas. It is not compatible with the -u flag.
-i pdb_file: **i**nput pdb file.

Significant effort was spent in ensuring that commands that require information not available will issue warning messages to that effect. In addition, online help is available using the "chop help *command*" syntax.

Summary of the commands recognized by chop:

Command	qualifier	qualifier	Arguments[a]
Open			Filename
Set	ligand		Residue_Spec
Add	ligand		Residue_Spec
Delete	ligand		Residue_Spec
Add	center		Atom_Spec
Set	cap	origin	Atom_Spec
Set	cap	size	Real
Set	cut	origin	*cap*
Set	cut	origin	*ligand*
Set	cut	origin	Atom_Spec
Set	cut	size	Real
Add	cut		Residue_Spec
Delete	cut		Residue_Spec
Set	mingap		Integer
Set	minchain		Integer
Set	neighborhood		Real
Set	threshold		Real
Fix	chains		
Cap			Residue_Spec
Cap	chain		Residue_Spec
Cap	all		
Uncap			Residue_Spec
Uncap	chain		Residue_Spec
Uncap	all		
Set	variable	origin	*cap*
Set	variable	origin	*cut*
Set	variable	origin	*ligand*
Set	variable	origin	Atom_Spec
Set	variable	size	Real
Add	variable		Residue_Spec
Delete	variable		Residue_Spec
Set	hid		Residue_Spec
Set	hie		Residue_Spec
Set	hip		Residue_Spec
Set	his		Residue_Spec
Neutralize			Residue_Spec
Neutralize	fixed		
Charge			Residue_Spec
Charge	fixed		
Set	targetq		Integer
Fix	charges		

Summary of the commands recognized by chop (continued):

Command	qualifier	qualifier	arguments[a]
set	view	default	
set	view	minimum	
set	view	full	
set	view	his	Integer
set	view		Residue_Spec
Status	ligand		
Status	cap		
Status	cut		
Status	variable		
Status	chains		
Status	charge		
Status	all		
Status			Residue_Spec
Write	pdb		Filename
Write	pepz	all	Filename
Write	pepz	variable	Filename
Write	translation		Filename

a: Atom_Spec is a full specification for an atom or group of atoms as recognized by midas, e. g. :asp@cd-*:glu@cg-* or :199a@ca. Note that you can graphically pick them in the screen or type them manually. Residue_Spec is the equivalent specification for residues. If it contains atom names they are ignored. Italicized items like *cap*, *cut*, and *ligand* are keywords that must be typed literally.

A detailed explanation of each command is given below, but a few general comments regarding syntax are needed. The commands are read one line at a time and are free format, the only requirement is that words are separated by at least one space. Although there is no order requirement per se, some commands may require data not provided yet, and will not cause any visible action until the appropriate information is entered by the appropriate command. In these cases an informational message to that respect will be given in the screen. The commands, qualifiers, and arguments are parsed independent of case and requires typing only enough characters to uniquely distinguish them, down to a minimum of three letters.

The commands that require arguments specified as Real or Integer in the table above should be given numbers of the corresponding types. Atom_Spec needs to be a full midas specification of an atom or group of atoms, with the exception of "set cap origin" in which case it should specify a single atom. Residue_Spec should be the full midas specification of a residue or group of residues, any atom name is ignored if given. The last two arguments can be either typed manually or "picked" through midas (use the <ALT> or <+> keys to start the picking mode. Italicized items like *cap*, *cut*, and *ligand* are keywords that must be typed literally.

• open Filename: reads the pdb file. This could also be accomplished by providing the command line argument "-i filename" when starting chop.

- set ligand Residue_Spec: this command and the next allow corrections to the (less than perfect) algorithm used to recognize the ligand.
- {add | delete} ligand Residue_Spec: adds or deletes the specified residues to the ligand set.

- add center Atom_Spec: adds a residue with a single "atom" at the geometric center of the specified atoms. The new residues and atoms are sequentially named C01, C02, etc. in the order they are created and can be used in any normal midas operation, but are not written in any of the files created.
- set cap origin Atom_Spec: adds a residue with a single "atom" at the same position as the atom selected. This new "atom" will be written in the pdb and pepz files created in the run. The input Atom_Spec can be a center defined by the command above.
- set cap size Real: adds a sphere of the given radius centered at the cap origin. It is useful to visualize the relative size of the system and the water cap to be used. The sphere may be deleted from the screen by using the command "set cap size 0.0" or using the midas command "~vdw :cap".

- set cut origin { Atom_Spec | *cap* | l*igand* }: sets the reference atoms used to select the residues to be included in the output files. The keywords *cap* and *ligand* can be used to avoid having to selectin the same sets of atoms again.
- set cut size Real: sets the minimum residue-residue distance beyond which residues are not included in the cut.
- {add | delete} cut Residue_Spec: add or delete the specified residues to the cut.

- fix chains: this command implements an automatic procedure that reduces the number of small, isolated chains generated by distance-based selections. It uses criteria based on the mingap, minchain, and threshold variables that can be modified using the three following commands.
- set mingap Integer: sets the minimum number residues allowed between two contiguous chains. The default is 3.
- set minchain Integer: sets the minimum number of residues allowed in a chain. The default is 2.
- set threshold Real: residues outside the threshold distance from the edge of the cut will not be deleted, *i. e.* the larger the threshold the more residues can be deleted. The default is 2.0.

- cap Residue_Spec: will convert the selected residue into the appropriate neutral "cap" acetyl or N-methylamide (ACE or AME) residue. It will not affect residues that are not at the termini of a chain.
- cap chain Residue_Spec: will cap both termini of the chain in which the selected residue is located.
- cap all: attempts to replace all termini of all chains by ACE or AME. If a condition exists that will not allow this operation it writes a diagnostic message without actually doing any changes.
- uncap {Residue_Spec | chain Residue_Spec | all }: will convert back the corresponding terminal residues to their original state in the input file.

- set variable origin { Atom_Spec | *cap* | *cut* | *ligand*}: sets the reference atoms used to select the residues to be included in the variable set. The keywords *cap*, *cut*, and *ligand* can be used to avoid having to select the same sets of atoms again.
- set variable size Real: sets the minimum residue-residue distance beyond which residues are not included in the variable set.

• {add | delete} variable Residue_Spec: add or delete the specified residues to the variable set.

• set { hid | hie | hip | his} Residue_Spec: changes the protonation or tautomeric state of the selected histidine residues.

• neutralize Residue_Spec: changes the specified residues to their neutral states, *e. g.* Asp to Ash, etc. It only affects Asp, Glu, Lys, Arg, and Hip residues.
• charge Residue_Spec: changes previously neutralized residues to their charged forms.
• neutralize *fixed*: will neutralize all the charged residues that have not included in the variable set. If this has not been defined it will write a message to that effect.
• charge *fixed*: reverses the effects of the command above.
• set targetq Integer: sets the target charge that the entire system must achieve. The default is 0.
• fix charges: will attempt to neutralize enough residues to achieve neutrality or the target charge specified by the command below. It uses a conservative algorithm based on the cap and the sizes of the cut and variable sets. It will not take any actions if all these have not been defined.

• set view default: will reset the midas view to the default state in terms of visibility, orientation, size, labels, and coloring. This is normally the C^{α}-trace for the protein, colored by atom type. The ligand carbon atoms are colored green, charged sidechains are displayed and colored red for Asp and Glu, blue for Lys, Arg, and Hip. The sidechain C's of His residues which have not been assigned an explicit protonation or tautomeric state are colored orange. Terminal residues are labelled. Residues not included in the current cut are shown in darker colors. The two commands below can modify this default.
• set view minimum: hides the residues not included in the cut. It is useful in cases of very large proteins that have very cluttered displays. This smaller view will become the default.
• set view full: reverses the effects of the previous command.
• set view his Integer: it allows the detailed examination of His residues. It clears the screen and then shows all the atoms of a histidine residue and all others residues within a residue-residue cutoff. The integer argument specifies the order of histidines currently selected as visible. *e. g.* "set view his 2" will show the second histidine included in the current set regardless of its real residue number. The residue-residue distance cutoff can be modified using the "set neighborhood" command.
• set view Residue_Spec: does the same as the command above for an arbitrary residue.
• set neighborhood Real: changes the residue-residue cutoffs used in the "set view" commands. The default is 4.5 Å.

• status { Residue_Spec | ligand | cap | cut | variable | chains charge | all }: gives some information in the screen about the item specified.

• write pdb Filename: writes a pdb file of the current selected cut that can be used as input to pepz. The residues and atoms are sequentially numbered starting at 1, chains are separated by TER records, and the residue names corresponding to any charge, protonation state, or capping modifications made will be included. The cap residue, if defined will be included.
• write pepz {all | variable} Filename: writes a template pepz input file that includes the correct sequence and numbering of the protein model. Other than the ligand, the only needed changes should be to include the correct names of files to be read and written. The "all"

keyword produces a pepz input file in which all degrees of freedom of all residues are variable, and is intended to produce a z-matrix for the conjugate gradient minimization. The "variable" keyword produces a pepz input in which all the backbone atoms are fixed, the sidechains of residues within the variable region are allowed to move, and the ligand has the same flexibility as the input boss Z-matrix.

• write translation Filename: writes a "translation table", *i. e.* a table that lists all the with their residue names, numbers and chain designators in the original pdb file along with the new residue names and numbers. Also included is an indication of whether they are variable or not and some of the distances to the cap, cut and variable references.

23 Appendix 3— Appendix 1 - User's guide to *clu*

The *clu* program (*Complex Ligand Utility*) included in the MCPRO distribution was originally designed to algorithmically replace a ligand within the Z-matrix of a complex. Over the course of its implementation, a few other capabilities were added in order to facilitate other repetitive and error prone operations encountered often in the course of biomolecular simulations. These capabilities include writing a PDB including all dummies, checking a PDB file, and extracting ligands from complexes.

Usage:

```
clu -t[options] target_file [-r[options] replacement_file] -n[options] new_file \
   [-o output_file] [-m method] [-v]

   Note that the function is independent of the order in which the flags and
   options are specified.

   -t target_file specifies the name of the target file that contains the structure
         to be operated on (i. e. the initial complex). It can be a Z-matrix or a
         PDB file, but a new Z-matrix can only be made if both input files are Z-
         matrices. The format of the input file can be determined automatically, but
         the user can specify it with the "f" option (see below).
   -r replacement_file specifies the file that contains the replacement ligand.
         Again, it can be a Z-matrix or a PDB, but if it is a PDB a Z-matrix can NOT
         be produced. Note that it is not required that this file contains an
         isolated ligand. If it is a complex, only the ligand (or the residues
         selected with the "s" option, see below) is replaced into the target. This
         can be used for swapping ligands between two complexes (in two separate
         passes of the program).
   -n new_file specifies the name of the new file to be created (i. e. the new
         complex)
   -o output_file (optional) is used to specify the name of a file to write the
         output. This will contain diagnostic and explanatory messages only.
   -m method to be used in the matching. See explanations below.
   -v if specified the output is much more verbose.


Options should be specified immediately after the flag without any blank spaces, and
      prefixed by a colon. Several options can be specified at once, as long as a
      colon precedes each option, i. e. "-t:f=z:s=159.a:e complex.z". The atom and
      residue specifications follow the syntax of the Midas/Chimera programs.

      :f=z|p  specifies the format of the file (Z-matrix or PDB). The format of the
            target and replacement files is automatically determined at run time if
            it is not specified.
      :m=atom_spec,atom_spec,... (only for target and replacement). Specifies
            selected atoms to fit using a rmsD scheme. This can be used to force a
```

unique superposition desired by the user. If the atom names are the same in the target and replacement structures they can be entered only once, for the first file specified.

:i=atom_spec,atom_spec,... (only for target and replacement). Specifies atoms that are to be **i**gnored during the matching procedure. Not that the atoms are retained and written to the new file if they are in the replacement file, they are only ignored during the fitting/matching.

:s=residue_spec,residue_spec,... (only target). This option can be used to select manually which residue(s) are to be treated as the ligand, this is normally determined automatically, but it can be overridden using this option. It is particularly useful if the ligand happens to be an aminoacid or a nucleic acid.

:c (no options) does only a check of the corresponding file. In this mode the output will contain the number of atoms, residues, chains, possible ligand identification, etc. In the case of a Z-matrix it will also give the number of variable and additional bonds, angles and dihedrals.

:e (only for target). Specifies that the ligand must be **e**xtracted from the complex target and written into the new Z-matrix or PDB file by itself. This can be useful both at the beginning of the setup of a new system in order to create the ligand Z-matrix using autozmat and BOSS, or to extract the Z-matrices of the unbound ligands from a series of complexes created by BOMB.

Methods: under normal usage, the ligand is replaced using an algorithm that attempts to maximize the overlap of heavy atoms giving preference to atoms of the same type. The user can specify one of several alternate methods using this option. Note that the first two methods follow a general rmsD fitting procedure between specific atoms or points in space rather than the more general matching procedure of the others.

fa **f**it **a**ll atoms using a rmsD procedure.
fc **f**it **c**enter of mass and the three atoms farthest from it.
ms **m**atch heavy atoms using a **s**imple scheme (the default).
mc **m**atch heavy atoms using a more **c**omplex scheme. In this case the algorithm tries to avoid the overlap of dissimilar atoms.
mq match charges. Presently this method can be used only if both target and replacement files are Z-matrices AND the charges are specified in them (i. e. they are "800 types" that are appended at the end of the file).

Examples:

clu -t:c 1fvv.pdb ► checks the input pdb file and outputs number of chains, the likely ligand identification, etc.

clu -t complex.z -n:f=p complex.pdb ► creates a pdb with the identical coordinates and position in space as the input z-matrix. Note that all the dummies in the Z-matrix are retained in the pdb written.

clu -t:e complex.pdb -n ligand.pdb ► extracts the ligand and writes the pdb formatted ligand.pdb file.

clu -t:e complex.z -n:f=z ligand.z ► extracts the ligand and writes a new z-matrix to the file ligand.z.

clu -t:s=nvp wt_nvp.z -r:s=MKC k103n_mkc.z -n:f=z wt_mkc.z
clu -r:s=nvp wt_nvp.z -t:s=MKC k103n_mkc.z -n:f=z k103n_nvp.z ► these two commands let you swap the ligands between the two z-matrices wt_nvp.z and k103n_mkc.z to create wt_mkc.z and k103n.z. Note that the only changes between the two commands are that the "-t" and "-r" flags were swapped, and the names of the output files.

clu -t:i=c12,c13,c14,c15,c16 complex.pdb -r new_ligand.z -n new_complex.pdb will create a pdb file of a new complex by matching the ligand in

"new_ligand.z" with that in the original complex. The atoms named C12, C13, C14, C15, and C16 in the original complex will be ignored during the matching procedure.

clu -t complex.z -r:i=c15,o06,n03,h22,h23 new_ligand.z -n:f=z new_complex.z
will create a Z-matrix file of a new complex by matching the ligand in "new_ligand.z" with that in the original complex. The atoms named C15, O06, N03, H22, H23 in the new ligand will be ignored during the the matching procedure, but they are still positioned correctly and written to the new file.

clu -t:s=lig complex.z -r new_ligand_conformer.z -m fa -n:f=z new_complex.z
will substitute a new ligand conformer into the target complex Z-matrix by doing an rmsD fit of all the atoms.

24 Appendix 3—Manually Positioning a Solvent Molecule

When MCPRO is used to solvate a new system, solvent molecules may not be automatically placed in all desired positions. For example, many HIV protease/drug complexes crystallize with a water molecule in the binding pocket, but MCPRO may determine that waters initially placed in that region make bad contacts with the protein or inhibitor and remove them. Often, a simulation of the system in which only the solvent may move will solve the problem, and a water molecule will move into the region of interest. If this does not happen, however, the in file from the simulation may be edited manually to position a solvent molecule as desired. The following prescription assumes the TIP3P water model is in use:

1. Read the MCPRO plt file and crystal structure (PDB) file into a graphics program such as MidasPlus and match the crystal structure file to the plt file orientation **without** changing the plt file orientation.

2. Find the coordinates of the crystallographic water molecule (usually only the oxygen atom) in this orientation.

3. Choose a water molecule from the plt file that is not interacting with a solute, *i.e.,* one from the edge of a solvent cap, and get its residue number (WTR#) and coordinates **without** changing the plt file orientation.

4. Calculate the desired coordinates for the water molecule's H-atoms (and the M atom of the TIP4P model if necessary); a simple translation of the molecule from its current position to the desired position:

Oxygen Atom Coordinates	Each Hydrogen Atom Coordinates
$X_{old} - X_{new} = \Delta X$	$X_{new} = X_{old} - \Delta X$
$Y_{old} - Y_{new} = \Delta Y$	$Y_{new} = Y_{old} - \Delta Y$
$Z_{old} - Z_{new} = \Delta Z$	$Z_{new} = Z_{old} - \Delta Z$

5. Now the new and old coordinates for the water molecule are known—search the in file for the old water coordinates and replace them with the new coordinates:

- Suppose there are RES solute residues and NMOL water molecules in the plt/in files.

- The water of interest WTR# is the (WTR# – RES)*th* water molecule in the list of solvent molecules.

- Replace the coordinates following the format of the in file below, excerpted from the subroutine INIT:

BASIC INFORMATION

```
   30 READ (IDSKI,470) NMOL,NSATM,NSOLUT,NVDIH,NVBND,NVANG,NACCPT,NRJECT
     *   ,IRN,IVERSN
      READ (IDSKI,480) TX,PX,EDG2
```

LAST ENERGIES

```
      READ (IDSKI,490) EOLD,EONOLD,ESONOL,ESOL1,ESOL2,EXXOLD,EXXOL1,
     *   EXXOL2,EDIHOL,EDIOL1,EDIOL2,ENBOL,ENBOL1,ENBOL2,EBCOLD,EBCOL1,
     *   EBCOL2,EBNDOL,EBNOL1,EBNOL2,EANGOL,EANOL1,EANOL2
      IF (IVERSN.GT.12) READ (IDSKI,490) ESONCO,ESONLO,EXXOLC,EXXOLL
```

SOLUTE COORDINATES: Reference, Pert1, Pert2

LAST VALUES of Variable Dihedrals, Bonds, Angles

```
      READ (IDSKI,500) ((ASOL(I,J),I=1,NSATM),J=1,3),((ASOL1(I,J),I=1,
     *   NSATM),J=1,3),((ASOL2(I,J),I=1,NSATM),J=1,3),(PHI(I),I=1,NVDIH)
     *   ,(BND(I),I=1,NVBND),(ANG(I),I=1,NVANG)
```

If more than 2 solutes (i.e., protein + drug + cap atom), then the RDEL and ADEL for Solute #3 are by default (good to search for in IN file):

```
   0.08000000   0.14000000
      IF (NSOLUT.GT.2) READ (IDSKI,500) (RDELS(I),I=3,NSOLUT),(ADELS(I),
     *   I=3,NSOLUT)
```

SOLVENT COORDINATES:

Format for TIP3P water molecule coordinates is six columns of data in the order:

```
Molecule1-O-Xcoor  Molecule2-O-Xcoor ...NMOL-O-Xcoor
Molecule1-O-Y      Molecule2-O-Y     ...NMOL-O-Y
Molecule1-O-Z      Molecule2-O-Z     ...NMOL-O-Z
Molecule1-H1-X     Molecule2-H1-X    ...NMOL-H1-X
Molecule1-H1-Y     Molecule2-H1-Y    ...NMOL-H1-Y
Molecule1-H1-Z     Molecule2-H1-Z    ...NMOL-H1-Z
Molecule1-H2-X     Molecule2-H2-X    ...NMOL-H2-X
Molecule1-H2-Y     Molecule2-H2-Y    ...NMOL-H2-Y
Molecule1-H2-Y     Molecule2-H2-Z    ...NMOL-H2-Z
```

(and similarly for point M in TIP4P)

6. Use this new in file to start a new simulation. It is recommended that a new plt file be generated quickly and checked to see that the water molecule is placed appropriately.

25 Appendix 4—Miscellaneous Notes for Users of MCPRO

Summary of OPLS–AA Atom Types for Amino Acids

Backbone entries are:

Atom	Type
amide C	235
amide O	236
Calpha in Gly	223
Calpha in Pro	246
Calpha in Ala, etc.	224
Calpha in Aib	225
H-Calpha in all AA	140
N sec-amide; all AA except Pro	238
N tert-amide; Pro	239
H-N in all AA	241

Side-chain entries are:

H on any saturated C is type 140
H on any sp2 C is type 146

AA	Atom	Type	AA	Atom	Type	AA	Atom	Type
Ala	CB	135	Aib	CB	135			
Val	CB	137	Val	CG	135			
Leu	CB	136	Leu	CG	137	Leu	CD	135
Ile	CB	137	Ile	CG1	135	Ile	CG2	136
Ile	CD2	135						
Pro	CB	136	Pro	CG	136	Pro	CD	245
Phe	CB	149	Phe	CG–CZ	145			
Ser	CB	157	Ser	OG	154	Ser	HO	155
Thr	CB	158	Thr	OG	154	Thr	HO	155
Thr	CG	135						
Tyr	CB	149	Tyr	CG–CE	145	Tyr	CZ	166
Tyr	O	167	Tyr	HO	168			
Trp	CB	136	Trp	CG	500	Trp	CD (CH)	514
Trp	CD (C)	501	Trp	NE	503	Trp	HN	504
Trp	CE (C)	502	Trp	other CH	145			
Cys	CB	206	Cys	SG	200	Cys	HS	204
Cyx	CB	214	Cyx	SG	203			
Met	CB	136	Met	CG	210	Met	SD	202
Met	CE	209						
Asp	CB	274	Asp	CG	271	Asp	OD	272
Glu	CB	136	Glu	CG	274	Glu	CD	271
Glu	OE	272						
Asn	CB	136	Asn	CG	235	Asn	OD	236
Asn	ND	237	Asn	HN	240			
Gln	CB	136	Gln	CG	136	Gln	CD	235
Gln	OE	236	Gln	NE	237	Gln	HN	240
Lys	CB–CD	136	Lys	CE	292	Lys	NZ	287
Lys	HN	290						
Arg	CB	136	Arg	CG	308	Arg	CD	307
Arg	NE	303	Arg	HNE	304	Arg	CZ	302
Arg	NN	300	Arg	HNN	301			
Hid	CB	505	Hid	CG	508	Hid	ND1	503
Hid	HND	504	Hid	CE1	506	Hid	HE	146
Hid	NE2	511	Hid	CD2	507	Hid	HD	146
Hie	CB	505	Hie	CG	507	Hie	ND1	511
Hie	CE1	506	Hie	HE	146	Hie	NE2	503
Hie	HNE	504	Hie	CD2	508	Hie	HD	146
Hip	CB	505	Hip	CG	510	Hip	ND1	512
Hip	HND	513	Hip	CE1	509	Hip	HE	146
Hip	NE2	512	Hip	HNE	513	Hip	CD2	510
Hip	HD	146						

Free Energy Perturbations with Full Flexibility

When a real atom goes to a dummy with full flexibility, one wants to be sure that at the end the dummy is not putting some constraints on the molecular geometry. So, in general, we make all torsion terms to the dummy zero (final type = 100) AND we set the force constants for angle bending for all but one angle to the dummy to zero. This requires entries in the .sb file like:

```
HC-CT-DM 0.0   109.47
```

We also often shrink the bond length to the dummy, but keep its force constant. This requires an entry in the .sb file like:

```
CA-DM 367.        0.30
```

This combination keeps the dummy in a reasonable position without constraining the final structure—if you do an optimization for this structure, you should get the same energy as from an optimization without the dummy.

Calculation of Absolute Free Energies of Hydration

Our first paper on computing absolute free energies of hydration by disappearing the solute is *Chem. Phys.* **129**, 193–200 (1989). *J. Comput. Chem*. **14**, 195–205 (1993) describes computing absolute free energies of hydration for substituted benzenes and deals with a system with torsional degrees of freedom.

For a rigid (note: rigid) molecule, one obtains the absolute free energy of hydration by making the molecule disappear in water. This is done by making the charges and Lennard-Jones parameters go to zero for each atom; we also reel in the bonds to *ca*. 0.1 Å. In the MCPRO Z-matrix file, the initial atom types would be for the real atoms and the final atom types would all be 100 ($q = \sigma = \varepsilon = 0.0$). Under geometrical variations, each atom from #2 on could be listed with parameter 1 (bond length) and final value of 0.1 (this says to shrink all bonds to 0.1 Å). A series of simulations would be run perturbing from RC (lambda) = 0.0 (real molecule) to 1.0 (all dummies); to avoid singularities, the last window should be run, for example, as RC0 = 0.90, RC1 = 0.80 and RC2 = 1.0, i.e. $0.80 \leftarrow 0.90 \rightarrow 1.0$, and not from $1.0 \rightarrow 0.90$.

A flexible molecule is trickier and different cases may be handled in different ways (see note above). Some examples are in the above *J. Comput. Chem*. paper. In general, one has to compute the free energy changes for making the molecule disappear both in the gas phase and in water.

Free Energy Perturbations: Changing RC Values *and* Reading from an IN File

This applies if you are running a simulation from an old IN file generated with different RC values, for example, if you have equilibrated a window with RC0=RC1=RC2 to save time and then changed to perturbation conditions for averaging, or if you are using the infile from one window to start the next one. In what follows, "`011` and `111` start" specify the ICALC, NEWRUN, and IPRNT values used in the command file. "IN/IN and ZIN/IN" refer to the solute and solvent origin specification used in the parameter file. GASP refers to the algorithm described in *J. Comput. Chem*. **16**, 311–327 (1995) and implemented in BOSS and MCPRO for perturbations of flexible degrees of freedom. The main idea is that the deviation from the equilibrium value of the reference state bond or angle is calculated for each perturbed degree of freedom (using parameters from the stretch-bend file). The perturbed state is generated by adding this difference to the current value of the variable bond or angle for the reference

(RC0) solute, resulting in the new geometries for the perturbed solutes. This is done for all flexible bonds and angles in the molecule.

Rigid perturbations: defined by geometry variations only

`011` start with IN/IN = reads and retains in file geometries for all solutes.

The Z-matrix is not rebuilt in a continuation run (ICALC = 0) and the geometry variation information is not used. This is normally *not* desirable.

`111` start with ZIN/IN = builds correct new geometries with geometry variations for all solutes.

The Z-matrix is regenerated, and geometry variations are incorporated. The last in file values for any dihedrals, angles, and bonds that *are* variable are used in the new structure. GASP is not implemented for rigid perturbations.

Flexible perturbations: no geometry variations, parameters drive the perturbation

`011` start with IN/IN = initial structures are the old geometries for all solutes.
`111` start with ZIN/IN = initial structures have old RC0 geometry, plus GASP perturbed geometries

In each case, solutes are rebuilt after the first solute move, so *running the simulation long enough will correct the initial geometries* and result in the same correct geometries based on GASP and the stretch-bend parameters.

Using geometry variations in a flexible perturbation:

We have tried using geometry variations to get close to the correct geometry quickly in a flexible perturbation of cyclohexane to benzene, where relying only on the parameters to update the reference structure requires long equilibration simulations in each window.

The results for `011` IN/IN and `111` ZIN/IN are the same as above for the flexible case. The geometry variations are ignored as expected for the continuation run (`011`), so only flexibility/GASP is used. For `111`, the Z-matrix is rebuilt with the geometry variations included, but since the dihedrals, etc. are *also* defined to be variable, they are overwritten with the last in file values of the variable dihedrals, angles, and bonds for RC0 and the resultant GASP geometries for RC1 and RC2.

In summary: Using a `111` start and ZIN/IN whenever an in file is read and RC values change does not hurt anything, and is *necessary* for rigid perturbations. One can use geometry variations for flexible degrees of freedom for ZMAT starts to try to set up the best initial geometry for a flexible perturbation, but there will be no advantage if an IN file geometry is read (*i.e.*, ZIN or IN for the solute origin).

Available Analysis Programs

bar –makes dihedral angle (and other) distribution plots from the "average" file generated with BOSS or MCPRO, suitable for display with the *xmgr* program.
HB15 –for hydrogen bond and interatomic distance analysis using the "save" file.

qrms –compares two "plt" files (PDB format) and calculates the root mean squared deviations for all residues.

Frequently Asked Questions

Q: *Is it possible to run MCPRO for only 1 configuration (MXCON = 1)? I just want to write a plt file to see if my new Z-matrix is read correctly.*

A: No—MXCON = 1 is not allowed; however, if you set MXCON = 2, the minimal amount of sampling will occur. If you have the simulation generate a solvent box or cap, and request only solvent moves (NSCHG = NSCHG2 = 999999), the solute conformation will be written out as it is read from the Z-matrix, without change.

Q: *I have been running simulations with a given Z-matrix, and now wish to start a new simulation, with an updated Z-matrix, from the coordinates of my last simulation. The variable-section of the old Z-matrix is fine, I just need the new internal coordinate representation. How can I get a new Z-matrix, without going through pepz again?*

A: Just declare NEWZMAT in the par file and the updated Z-matrix is appended to the sum file. Alternatively, one could use the *pepz* utility program *pdb2newzm*. It requires the old Z-matrix, the name for the new Z-matrix, and a PDB file with the desired atomic coordinates corresponding to the old Z-matrix (*ie*., the last plt file from the original simulation).

Q: *I have used type –1 dummy atoms in my Z-matrix to define the position of a ligand, and they are not written to the plt file. How can I see where they may have moved during a simulation as a result of rigid-body ligand moves?*

A: Type 100 dummy atoms **will** be written to the plt file. You could run a quick simulation to generate a new plt file, after changing the type –1 atoms to type 100 in the Z-matrix. If you want to be certain that the solute configuration remains unchanged in this additional simulation, set it up for solvent-only moves (NSCHG = NSCHG2 = 999999).

Q: *I would like to make minor changes to a Z-matrix, but want to use the files from the original simulation to start the next one. How can I use the information, especially the now well-equilibrated solvent cap, from an old IN file? I know that an old IN file cannot be read directly if the Z-matrix has changed.*

A: You may change non-bonded or torsional parameters or dihedral angle variations in a Z-matrix and still use an old in file to start the new simulation (`011` start, as to restart averaging of energies).

For other modifications, you may wish to use *pdb2newzm* to build a Z-matrix with the internal coordinates from the last plt file of the simulation (see above). This Z-matrix will then have the same solute orientation relative to the solvent cap as the solute information in the last IN file. Modify the Z-matrix as needed; delete atoms from the solute, or remove or add variable bonds, for example. Now the only difficulty with using the IN file is that some atomic coordinates and/or the variable dihedral, angle, and bond values in the file may be in the wrong order relative to your new Z-matrix, or may be missing, if new atoms or variables have been added. The trick

is to generate a new IN file with the correct solute information, then cut-and-paste the old solvent coordinate information you need to finish the new IN file and start the next simulation.

1) Run a MXCON = 2, solvent-only (NSCHG = NSCHG2 = 999999) simulation, generating a temporary water cap around your new Z-matrix solute(s). Follow the same method as you used to build a cap in your original simulation. The resulting IN file will contain the header, solute coordinates and values for any variable dihedrals, angles, and bonds associated with the new Z-matrix, and a new water cap.

2) Search through the new IN file to the beginning of the water section. Delete from there to the end of the file. (See Appendix 2 and subroutine INIT for the format of the IN file.)

3) Search through the old IN file to the analogous place, and copy and paste from there to the end of that file into the new IN file to replace the water coordinates.

4) Use this modified new IN file to run another 2-configuration solvent-only simulation, with a `011` start. Compare the resulting plt file with the last plt file of the old simulation to check that only 1–2 water molecules have moved and that the orientation of the solute(s) and solvent are correct. If so, you may safely start the next simulation with your new Z-matrix and IN files.

26 Appendix 5 - OPLS-AA Force Field for Organic and Biomolecular Systems

The parameters for the force field are distributed with the BOSS and MCPRO programs in the files *oplsaa.par* and *oplsaa.sb*.

Form of the Force Field

Bond stretching: $$E_{bond} = \sum_{bonds} K_r (r - r_{eq})^2$$

Angle bending: $$E_{angle} = \sum_{angles} K_\theta (\theta - \theta_{eq})^2$$

Torsion: $$E_{dih} = V_1(1+\cos\phi)/2 + V_2(1-\cos 2\phi)/2 + V_3(1+\cos 3\phi)/2 + V_4(1-\cos 4\phi)/2$$

Non-bonded: $$E_{ab} = \sum_{i}^{on\ a} \sum_{j}^{on\ b} [q_i q_j e^2 / r_{ij} + 4\varepsilon_{ij}(\sigma_{ij}^{12} / r_{ij}^{12} - \sigma_{ij}^{6} / r_{ij}^{6})] f_{ij}$$

f_{ij} = 0.5 if i, j are 1,4; otherwise, f_{ij} = 1.0

References

W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, *J. Am. Chem. Soc.* **118**, 11225-11236 (1996).

W. Damm, A. Frontera, J. Tirado-Rives, and W. L. Jorgensen, *J. Comput. Chem.* **18**, 1955-1970 (1997).

W. L. Jorgensen and N. A. McDonald, *Theochem*. **424**, 145-155 (1998).

W. L. Jorgensen and N. A. McDonald, *J. Phys. Chem. B* **102**, 8049-8059 (1998).

R. C. Rizzo and W. L. Jorgensen, *J. Am. Chem. Soc.* **121**, 4827-4836 (1999).

E. K. Watkins and W. L. Jorgensen, *J. Phys. Chem. A* **105**, 4118-4125 (2001).

M. L. P. Price, D. Ostrovsky, and W. L. Jorgensen, *J. Comput. Chem.* **22**, 1340-1352 (2001).

W. L. Jorgensen, J. P. Ulmschneider, J. Tirado-Rives, J. Phys. Chem. B 108, 16264-70 (2004).

K. P. Jensen and W. L. Jorgensen, J. Chem. Theory Comput. 2, 1499-1509 (2006).

```
*****************************************************************
OPLS-AA atom types for bond stretching, angle bending and torsions.
*****************************************************************

AMBER all-atom definitions are used (JACS 117, 5179 (1995), see next page)
with the following additions:

CO       acetal C  (O- CR2 -O)
CM       alkene C
C=       C2 in dienes (C1 is CM)
C#       C3 in trienes (C1 is CM; C2 is C=)
CZ       sp C in triple bond
C+       carbenium carbon
C!       C1 in biphenyl-like ring
CS       C2 in heterocycle 5-ring like pyrrole, furan, thiophene
CU       C next to N:-X in aromatic 5-ring like pyrazole, isoxazole
CX       C in C=C in protonated imidazole, histidine
CY       sp3 carbon in 3-ring or 4-ring
C$       carbonyl carbon in 3-ring or 4-ring (e.g, lactam)
C:       allene or ketene C2

N$       amide N in 3-ring or 4-ring
NZ       sp N in triple bond or azide
NT       amine, aniline N
NO       N in N=O (nitro, nitroso)
N=       N2 in azadiene C=N-C=C, etc.
NM       N in tertiary amide
from AMBER: N, NA, NB, NC, N2, N3, NY, N*

ON       O in N=O
OY       O in S=O
O$       O in 3-ring or 4-ring, e.g., epoxide

P+       phosphonium P

SY       S in S=O in sulfonamide or sulfone
SZ       S in sulfoxide
S=       S in thiocarbonyl (thiourea, thioamide,...)
```

Table 1. List of Atom Types[a]

atom	type	description
carbon	CT	any sp^3 carbon
	C	any carbonyl sp^2 carbon
	CA	any aromatic sp^2 carbon and (Cϵ of Arg)
	CM	any sp^2 carbon, double bonded
	CC	sp^2 aromatic in 5-membered ring with one substituent + next to nitrogen (Cγ in His)
	CV	sp^2 aromatic in 5-membered ring next to carbon and lone pair nitrogen (e.g. Cδ in His (δ))
	CW	sp^2 aromatic in 5-membered ring next to carbon and NH (e.g. Cδ in His (ϵ) and in Trp)
	CR	sp^2 aromatic in 5-membered ring next to two nitrogens (Cγ and Cϵ in His)
	CB	sp^2 aromatic at junction of 5- and 6-membered rings (Cδ in Trp) and both junction atoms in Ade and Gua
	C*	sp^2 aromatic in 5-membered ring next to two carbons (e.g. Cγ in Trp)
	CN	sp^2 junction between 5- and 6-membered rings and bonded to CH and NH (Cϵ in Trp)
	CK	sp^2 carbon in 5-membered aromatic between N and N-R (C8 in purines)
	CQ	sp^2 carbon in 6-membered ring between lone pair nitrogens (e.g. C2 in purines)
nitrogen	N	sp^2 nitrogen in amides
	NA	sp^2 nitrogen in aromatic rings with hydrogen attached (e.g. protonated His, Gua, Trp)
	NB	sp^2 nitrogen in 5-membered ring with lone pair (e.g. N7 in purines)
	NC	sp^2 nitrogen in 6-membered ring with lone pair (e.g. N3 in purines)
	N*	sp^2 nitrogen in 5-membered ring with carbon substituent (in purine nucleosides)
	N2	sp^2 nitrogen of aromatic amines and guanidinium ions
	N3	sp^3 nitrogen
oxygen	OW	sp^3 oxygen in TIP3P water
	OH	sp^3 oxygen in alcohols, tyrosine, and protonated carboxylic acids
	OS	sp^3 oxygen in ethers
	O	sp^2 oxygen in amides
	O2	sp^2 oxygen in anionic acids
sulfur	S	sulfur in methionine and cysteine
	SH	sulfur in cysteine
phosphorus	P	phosphorus in phosphates
hydrogen	H	H attached to N
	HW	H in TIP3P water
	HO	H in alcohols and acids
	HS	H attached to sulfur
	HA	H attached to aromatic carbon
	HC	H attached to aliphatic carbon with no electron-withdrawing substituents
	H1	H attached to aliphatic carbon with one electron-withdrawing substituent
	H2	H attached to aliphatic carbon with two electron-withdrawing substituents
	H3	H attached to aliphatic carbon with three electron-withdrawing substituents
	HP	H attached to carbon directly bonded to formally positive atoms (e.g. C next to NH_3^+ of lysine)
	H4	H attached to aromatic carbon with one electronegative neighbor (e.g. hydrogen on C5 of Trp, C6 of Thy)
	H5	H attached to aromatic carbon with two electronegative neighbors (e.g. H8 of Ade and Gua and H2 of Ade)