# GRASP$^2$ manual

Patrick Norrington

January 24, 2002

GRASP has been developed by Prof Ian Grant and his collaborators. Any inquiries or comments on the package can be directed to Dr Norrington at

Department of Applied Mathematics and Theoretical Physics
The Queen's University of Belfast
Belfast
Northern Ireland
BT7 1NN
United Kingdom

telephone : +44 (0) 28 9027 3190
fax : +44 (0) 28 9023 9182

email : p.norrington@qub.ac.uk
homepage : http://www.am.qub.ac.uk/users/p.norrington

**DISCLAIMER**

These codes have been developed in good faith. Mostly they produce reliable results. However the user is warned to be always skeptical with anything numerical.

# Contents

# Chapter 1

# Theory

# Chapter 2

# The MCDF method

In the following sections we present an overview of the multi-configuration Dirac-Fock method (MCDF) approximation for the calculation of atomic states, level energies and transition data. Unless indicated Hartree atomic units will be used throughout. This theory is the basis of the GRASP package [10].

## 2.1 Relativistic orbitals

A relativistic (or Dirac) orbital $|n\kappa m\rangle$ is an eigenfunction of the angular momentum operators $\hat{j}^2$ and $\hat{j}_z$ where $\hat{\boldsymbol{j}} = \hat{\boldsymbol{l}} + \hat{\boldsymbol{s}}$, (sum of orbital and spin) i.e.

$$\hat{j}^2 \, |n\kappa m\rangle = j(j+1) \, |n\kappa m\rangle \qquad \text{with } j = \frac{1}{2}, \frac{3}{2}, \cdots \tag{2.1}$$

$$\hat{j}_z \, |n\kappa m\rangle = m \, |n\kappa m\rangle \qquad \text{with } m = -j, \cdots, j \tag{2.2}$$

and of the relativistic parity operator $\hat{p} = \beta\hat{\pi}$ ($\hat{\pi}$ is the usual parity operator and the Dirac matrix $\beta$ is defined below)

$$\hat{p} \, |n\kappa m\rangle = (-)^l \, |n\kappa m\rangle \tag{2.3}$$

The principal quantum number is $n = 1, 2, \cdots$ and $\kappa = \mp 1, \mp 2, \cdots$ is the relativistic quantum number.

$\kappa$ is given by $\kappa = \pm(j + \frac{1}{2})$ for $l = j \pm \frac{1}{2}$. Thus

$$j = |\kappa| - \tfrac{1}{2} \qquad \text{and} \qquad l = \begin{cases} |\kappa| - 1 & \text{for } \kappa < 0 \\ \kappa & \text{for } \kappa > 0 \end{cases} \tag{2.4}$$

The parity is determined only by the orbital angular momentum $l = 0, 1, \cdots$.

| spectroscopic label | s | $\overline{\text{p}}$ | p | $\overline{\text{d}}$ | d | $\overline{\text{f}}$ | f | $\overline{\text{g}}$ | g |
|---|---|---|---|---|---|---|---|---|---|
| $l$ | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 |
| $j$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{5}{2}$ | $\frac{5}{2}$ | $\frac{7}{2}$ | $\frac{7}{2}$ | $\frac{9}{2}$ |
| $\kappa$ | $-1$ | 1 | $-2$ | 2 | $-3$ | 3 | $-4$ | 4 | $-5$ |
| parity | $e$ | $o$ | $o$ | $e$ | $e$ | $o$ | $o$ | $e$ | $e$ |

Each of the $\leq 2j+1$ orbitals with the same $(n\kappa)$ but differing $m$ quantum numbers (referred to as a **subshell**) are assumed to have the same radial form. Using the convention of Grant [1] an explicit representation is

$$\langle \boldsymbol{r}|n\kappa m\rangle = \frac{1}{r}\begin{pmatrix} P_{n\kappa}(r)\,\chi^m_\kappa(\theta,\phi) \\ iQ_{n\kappa}(r)\,\chi^m_{-\kappa}(\theta,\phi)\end{pmatrix} \tag{2.5}$$

Here $P_{n\kappa}(r)$ and $Q_{n\kappa}(r)$ are respectively the large and small component radial wavefunctions, and the functions $\chi^m_\kappa(\theta,\phi)$ are the spinor spherical harmonics

$$\chi^m_\kappa(\theta,\phi) = \sum_{\sigma=\pm\frac{1}{2}} \langle l, m-\sigma, \tfrac{1}{2}\sigma|l\tfrac{1}{2}jm\rangle\; Y_l^{m-\sigma}(\theta,\phi)\;\chi^\sigma \tag{2.6}$$

where $\langle l, m-\sigma, \frac{1}{2}\sigma|l\frac{1}{2}jm\rangle$ is a Clebsch-Gordan coefficient, $Y_l^{m-\sigma}(\theta,\phi)$ is a spherical harmonic and $\chi^\sigma$ is a spinor basis function.

The spinor spherical harmonics satisfy

$$\hat{j}^2\;\chi^m_\kappa(\theta,\phi) = j(j+1)\;\chi^m_\kappa(\theta,\phi) \tag{2.7}$$

$$\hat{j}_z\;\chi^m_\kappa(\theta,\phi) = m\;\chi^m_\kappa(\theta,\phi) \tag{2.8}$$

$$\hat{l}^2\;\chi^m_\kappa(\theta,\phi) = l(l+1)\;\chi^m_\kappa(\theta,\phi) \tag{2.9}$$

$$\hat{s}^2\;\chi^m_\kappa(\theta,\phi) = \tfrac{3}{4}\;\chi^m_\kappa(\theta,\phi) \tag{2.10}$$

$$\hat{\pi}\;\chi^m_\kappa(\theta,\phi) = (-)^l\;\chi^m_\kappa(\theta,\phi) \tag{2.11}$$

The angular momentum algebra is simplest when the orbitals are chosen to form an orthonormal set

$$\langle n_a\kappa_a m_a|n_b\kappa_b m_b\rangle = \delta_{ab} \tag{2.12}$$

whence it is necessary to impose

$$N(ab) = \begin{cases} 0 & \text{when } a\neq b \text{ but } \kappa_a = \kappa_b \\ 1 & \text{when } a = b\end{cases} \tag{2.13}$$

and

$$N(ab) = \int_0^\infty dr\;\left(P_a(r)P_b(r) + Q_a(r)Q_b(r)\right) \tag{2.14}$$

It is often advantageous to allow for a limited amount of nonorthogonality [8,9] but this will not be discussed here.

## 2.2 Configuration state functions

A **configuration state function** (CSF), $|\gamma PJM\rangle$, of an $N$-electron system is formed by taking linear combinations of Slater determinants of order $N$ constructed from the Dirac orbitals so as to obtain normalised

$$\langle \gamma PJM|\gamma PJM\rangle = 1 \tag{2.15}$$

eigenfunctions of the parity operator $\hat{P}$ and total angular momentum operators $\hat{J}^2$ and $\hat{J}_z$

$$\hat{P}\,|\gamma PJM\rangle = P\,|\gamma PJM\rangle \tag{2.16}$$

$$\hat{J}^2\,|\gamma PJM\rangle = J(J+1)\,|\gamma PJM\rangle \tag{2.17}$$

$$\hat{J}_z\,|\gamma PJM\rangle = M\,|\gamma PJM\rangle \qquad \text{with } M = -J, \cdots, J \tag{2.18}$$

The label $\gamma$ represents all information such as orbital occupation numbers, coupling, seniority numbers, etc. , required to define the CSF uniquely.

The **standard** coupling scheme for a CSF is defined as follows.

- Firstly the electrons are assigned to subshells by specifying the orbital occupation numbers, $q(a) \leq 2j_a + 1$. For each subshell $a$ the electrons are $jj$-coupled to give a seniority $\nu_a$ and angular momentum $J_a M_a$

  $$|(j_a)^{q(a)}\nu_a J_a M_a\rangle \tag{2.19}$$

  The following tables list the allowed values of the quantum numbers $\nu$ and $J$ for a given $j^q$.

| $j$ | $q$ | $\nu$ | $J$ |
|---|---|---|---|
| $\frac{1}{2}$ | 0, 2 | 0 | 0 |
| | 1 | 1 | $\frac{1}{2}$ |
| $\frac{3}{2}$ | 0, 4 | 0 | 0 |
| | 1, 3 | 1 | $\frac{3}{2}$ |
| | 2 | 0 | 0 |
| | | 2 | 2 |
| $\frac{5}{2}$ | 0, 6 | 0 | 0 |
| | 1, 5 | 1 | $\frac{5}{2}$ |
| | 2, 4 | 0 | 0 |
| | | 2 | 2, 4 |
| | 3 | 1 | $\frac{5}{2}$ |
| | | 3 | $\frac{3}{2}$, $\frac{9}{2}$ |
| $\frac{7}{2}$ | 0, 8 | 0 | 0 |
| | 1, 7 | 1 | $\frac{7}{2}$ |
| | 2, 6 | 0 | 0 |
| | | 2 | 2, 4, 6 |
| | 3, 5 | 1 | $\frac{7}{2}$ |
| | | 3 | $\frac{3}{2}$, $\frac{5}{2}$, $\frac{9}{2}$, $\frac{11}{2}$, $\frac{15}{2}$ |
| | 4 | 0 | 0 |
| | | 2 | 2, 4, 6 |
| | | 4 | 2, 4, 5, 8 |

| $j$ | $q$ | $\nu$ | $J$ |
|---|---|---|---|
| $\frac{9}{2}$ | 0, 10 | 0 | 0 |
| | 1, 9 | 1 | $\frac{9}{2}$ |
| | 2, 8 | 0 | 0 |
| | | 2 | 2, 4, 6, 8 |
| $\frac{11}{2}$ | 0, 12 | 0 | 0 |
| | 1, 11 | 1 | $\frac{11}{2}$ |
| | 2, 10 | 0 | 0 |
| | | 2 | 2, 4, 6, 8, 10 |
| $\frac{13}{2}$ | 0, 14 | 0 | 0 |
| | 1, 13 | 1 | $\frac{13}{2}$ |
| | 2, 12 | 0 | 0 |
| | | 2 | 2, 4, 6, 8, 10, 12 |
| $\frac{15}{2}$ | 0, 16 | 0 | 0 |
| | 1, 15 | 1 | $\frac{15}{2}$ |
| | 2, 14 | 0 | 0 |
| | | 2 | 2, 4, 6, 8, 10, 12, 14 |

Extra group-theoretic labels may be required to specify the state if the number of electrons in a subshell having $j \geq 9/2$ satisfies $2 < q(a) < 2j_a - 1$. Because tables of $jj$-coupling coefficients of fractional parentage have only been provided for $j \leq 9/2$ in the CFP section of GRASP [10], such configurations are not discussed here.

- Next subshell angular momentum $J_1$ and $J_2$ are coupled to give an intermediate momentum $X_1$ which in turn is coupled to $J_3$ to give an $X_2$ and so on until all subshells have been coupled to give a total angular momentum $J$

$$(\cdots((J_1J_2)X_1J_3)X_2\cdots)J \tag{2.20}$$

CSFs formed by redistributing electrons among the subshells and changing the coupling sequence are orthogonal.

Sometimes it is useful to express CSFs in a coupling scheme other than the standard $jj$-coupling, especially if it gives a more clear-cut classification of the atomic states. Whilst the internal representation of GRASP is always as described above, a facility (an adaption of the TRANSFORM package of Dyall [11]) is provided to enable the user to express the reference CSFs and print the final results in **nonstandard** $LS$- or $jl$-coupled bases.

There are two program limitations to note in this context. The first is that only up to four open $(nl)^q$ shells can be handled by the program at present; this can be extended. The second restriction is imposed by the tabulated $jj$-$LS$ transformation data which limits the number of electrons or holes in open shells with $l > 2$ to not more than 2.

## 2.3 Atomic state functions

An **atomic state function** (ASF) is a linear combination of CSFs sharing common values of $P$, $J$ and $M$

$$|\Gamma PJM\rangle = \sum_{r=1}^{n_c} c_{r\Gamma}\,|\gamma_r PJM\rangle \tag{2.21}$$

The **mixing coefficients** $c_{r\Gamma}$ may be combined in a column vector $\boldsymbol{c}_\Gamma \equiv \{c_{r\Gamma}, r = 1\cdots n_c\}$ This is the representation of the atomic state $|\Gamma PJM\rangle$ with respect to the CSF basis set $\{|\gamma_r PJM\rangle, r = 1\cdots n_c\}$

The ASF will be chosen to be orthonormal so that

$$(\boldsymbol{c}_{\Gamma_i})^\dagger \boldsymbol{c}_{\Gamma_j} = \delta_{ij} \tag{2.22}$$

where dagger denotes the Hermitian conjugate.

## 2.4 The Dirac-Coulomb hamiltonian

All of the dominant interactions in an $N$-electron atom or ion are included in the Dirac-Coulomb hamiltonian

$$\hat{H}^{DC} = \sum_{i=1}^{N} \hat{H}_i + \sum_{i=1}^{N}\sum_{j=i+1}^{N} |\boldsymbol{r}_i - \boldsymbol{r}_j|^{-1} \tag{2.23}$$

The first term on the rhs

$$\hat{H} = c\sum_{i=1}^{3} \alpha_i \hat{p}_i + (\beta - 1)c^2 + V_{nuc}(r) \tag{2.24}$$

is the one-body contribution for an electron due to its kinetic energy and interaction with the nucleus — the rest energy $c^2$ has been subtracted out. The nuclear potential $V_{nuc}(r)$ takes the Coulomb form $-Z/r$ (where $Z$ is the atomic number of the system) when nuclear volume effects are neglected.

In the **standard representation** the $4 \times 4$ **Dirac matrices** $\alpha_i, \beta$ are in partitioned form

$$\alpha_i = \begin{pmatrix} 0 & \sigma_i \\ \sigma_i & 0 \end{pmatrix} \qquad \beta = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.25}$$

where $i = 1, 2, 3$ and $\sigma_i$ are the usual $2 \times 2$ Pauli matrices.

The two-body instantaneous Coulomb interactions between the electrons comprise the second term.

Higher order (QED) modifications to the hamiltonian are due to the transverse electromagnetic interaction and the radiative corrections. These are treated by perturbation theory.

The matrix of $\hat{H}^{DC}$ with respect to a basis of CSFs plays a central role in all relativistic atomic structure calculations. Using the CSF expansion of atomic state $\Gamma$ gives an approximation to its energy as

$$\begin{aligned} E_\Gamma^{DC} &= \langle \Gamma PJM | \hat{H}^{DC} | \Gamma PJM \rangle \\ &= (\boldsymbol{c}_\Gamma^{DC})^\dagger \boldsymbol{H}^{DC} \boldsymbol{c}_\Gamma^{DC} \end{aligned} \tag{2.26}$$

The hamiltonian matrix $\boldsymbol{H}^{DC}$ has the elements

$$H_{rs}^{DC} = \langle \gamma_r PJM | \hat{H}^{DC} | \gamma_s PJM \rangle \tag{2.27}$$

Requiring $E_\Gamma^{DC}$ to be stationary with respect to variations in the mixing coefficients subject to eq.(2.22) leads to the eigenvalue problem for the mixing coefficients

$$(\boldsymbol{H}^{DC} - E_\Gamma^{DC} \boldsymbol{I})\ \boldsymbol{c}_\Gamma^{DC} = 0 \tag{2.28}$$

Here $\boldsymbol{I}$ is the $n_c \times n_c$ unit matrix.

## 2.5 The hamiltonian matrix

The hamiltonian matrix elements can be expressed in terms of angular coefficients and radial integrals (see Grant [1]).

One-body interactions give rise to the $I(ab)$ integrals

$$\begin{aligned} I(ab) &= \delta_{\kappa_a \kappa_b} \int_0^\infty dr \,\big[ c(Q_a(r)P_b'(r) - P_a(r)Q_b'(r)) \\ &\quad -2c^2\, Q_a(r)Q_b(r) + \frac{c\kappa_b}{r}\,(P_a(r)Q_b(r) + Q_a(r)P_b(r)) \\ &\quad V_{nuc}(r)\,(P_a(r)P_b(r) + Q_a(r)Q_b(r))\big] \end{aligned} \tag{2.29}$$

where $f' = df/dr$.

Two-body interactions yield relativistic Slater integrals

$$R^k(abcd) = \int_0^\infty dr \left[ (P_a(r)P_c(r) + Q_a(r)Q_c(r)) \frac{1}{r} Y^k(bd;r) \right] \tag{2.30}$$

and the relativistic Hartree $Y$-functions are defined by

$$Y^k(bd;r) = r \int_0^\infty ds\, U^k(r,s)\, (P_b(s)P_d(s) + Q_b(s)Q_d(s)) \tag{2.31}$$

where

$$U^k(r,s) = \begin{cases} r^k/s^{k+1} & \text{if } r < s \\ s^k/r^{k+1} & \text{if } s < r \end{cases} \tag{2.32}$$

A diagonal contribution to the hamiltonian matrix can be written

$$\begin{aligned} H^{DC}_{rr} &= \sum_{a=1}^{n_o} \left( q_r(a) I(aa) + \sum_{b \geq a}^{n_o} \sum_{k=0,2,\cdots}^{k_0} f^k_r(ab)\, F^k(ab) \right. \\ &+ \left. \sum_{b>a}^{n_o} \sum_{k=k_1,k_1+2,\cdots}^{k_2} g^k_r(ab)\, G^k(ab) \right) \end{aligned} \tag{2.33}$$

where $n_o$ is the number of orbitals. In this expression $F$ and $G$ are **direct** and **exchange** radial integrals

$$F^k(ab) = R^k(abab) \qquad G^k(ab) = R^k(abba) \tag{2.34}$$

$q_r(a)$ is the occupation number of orbital $a$ in the CSF $r$. The limits $k_0$, $k_1$ and $k_2$ are

$$k_0 = (2j_a - 1)\, \delta_{ab} \tag{2.35}$$

$$k_1 = \begin{cases} |j_a - j_b| & \text{if } \kappa_a\kappa_b > 0 \\ |j_a - j_b| + 1 & \text{if } \kappa_a\kappa_b < 0 \end{cases} \tag{2.36}$$

$$k_2 = \begin{cases} j_a + j_b & \text{if } j_a + j_b - k_1 \text{ is even} \\ j_a + j_b - 1 & \text{otherwise} \end{cases} \tag{2.37}$$

The angular coefficients $f^k_r(ab)$ and $g^k_r(ab)$ have the form

$$f^0_r(aa) = \tfrac{1}{2} q_r(a)(q_r(a) - 1) \qquad \text{and} \qquad f^0_r(ab) = q_r(a)q_r(b) \tag{2.38}$$

For $k > 0$ and a full subshell ( i.e. , $q_r(a) = 2j_a + 1$ or $q_r(b) = 2j_b + 1$)

$$\begin{aligned} f^k_r(ab) &= -\tfrac{1}{2} \left[ q_r(a) C(a,k,a) \right]^2 \delta_{ab} \\ g^k_r(ab) &= -q_r(a)q_r(b) \left[ C(a,k,b) \right]^2 \\ C(a,k,b) &= \begin{pmatrix} j_a & k & j_b \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{pmatrix} \end{aligned} \tag{2.39}$$

For $k > 0$ and both subshells open ( i.e. , $q_r(a) < 2j_a + 1$ and $q_r(b) < 2j_b + 1$)

$$\begin{aligned} f_r^k(ab) &= V_{rr}^k(abab) \\ g_r^k(ab) &= V_{rr}^k(abba) \end{aligned} \tag{2.40}$$

For off-diagonal ($r \neq s$)

$$H_{rs}^{DC} = \sum_{abcd} \sum_{k} V_{rs}^k(abcd)\, R^k(abcd) + \sum_{ab} T_{rs}(ab)\, I(ab) \tag{2.41}$$

The **configuration coupling coefficients** $V_{rs}^k(abcd)$ and $T_{rs}(ab)$ are discussed later.

## 2.6 Generation of the radial functions

A pair of bound-state radial wavefunctions $P_a$ and $Q_a$ (which depend on $n_a\kappa_a$) for a subshell $a$ may be obtained, in general, by solving a pair of radial Dirac equations

$$\left(\frac{d}{dr} + \frac{\kappa_a}{r}\right) P_a(r) - \left(2c - \frac{\epsilon_a}{c} + \frac{Y_a(r)}{cr}\right) Q_a(r) = -\frac{X_a^{(P)}(r)}{r} \tag{2.42}$$

$$\left(\frac{d}{dr} - \frac{\kappa_a}{r}\right) Q_a(r) + \left(-\frac{\epsilon_a}{c} + \frac{Y_a(r)}{cr}\right) P_a(r) = \frac{X_a^{(Q)}(r)}{r}$$

with $\epsilon_a > 0$ and subject to the boundary conditions

$$\left.\begin{matrix} P_a \\ Q_a \end{matrix}\right\} = 0 \qquad \text{when } r = 0 \tag{2.43}$$

$$\left.\begin{matrix} P_a \\ Q_a \end{matrix}\right\} \to 0 \qquad \text{when } r \to \infty$$

$$P_a' > 0 \qquad \text{when } r \to 0$$

and the orthonormalisation condition eq.(2.7). The asymptotic forms of, and the relation between, the large- and small-component functions near the origin are dependent upon the behaviour of the potential energy function $Y_a(r)$ which dominates the inhomogeneous terms $X_a^{(P)}(r)$ and $X_a^{(Q)}(r)$ in this neighbourhood [3].

These equations along with their boundary conditions define an **eigenvalue problem** for the orbitals $P_a$ and $Q_a$ and the energies $\epsilon_a$ ($a = 1 \cdots n_o$) when

1. $Y_a(r) = Y_{\kappa_a}(r)$ i.e. , the orbitals with the same angular quantum number $\kappa$ are generated in the same potential

2. $X_a^{(P)}(r)$ and $X_a^{(Q)}(r)$ are both zero.

We treat three simple but important cases

1. When $Y_a(r) = Z$ we have the Coulomb central field and the solutions may be obtained analytically [12]

$$\left.\begin{array}{c} P_{n\kappa}(r) \\ Q_{n\kappa}(r) \end{array}\right\} = -S_\kappa \mathcal{N} \sqrt{1 \pm E}\, (\Phi_1(\rho) \pm \Phi_2(\rho))\, \rho^\gamma\, \mathrm{e}^{-\frac{\rho}{2}} \tag{2.44}$$

where

- $$\rho = \tfrac{2Zr}{N} \tag{2.45}$$

- $$\gamma = \sqrt{\kappa^2 - (\alpha Z)^2} < |\kappa| \tag{2.46}$$

- $\mathcal{N}$ is a normalisation constant given by

$$\mathcal{N} = \left[ \left( \frac{Z}{2N^2(N-\kappa)} \right) \frac{\Gamma(2\gamma + n_r + 1)}{n_r!} \right]^{\frac{1}{2}} \frac{1}{\Gamma(2\gamma+1)} \tag{2.47}$$

- $\Phi_1$ and $\Phi_2$ are polynomials in $\rho$ given by

$$\begin{aligned} \Phi_1(\rho) &= -n_r\, M(-n_r + 1, 2\gamma + 1, \rho) \\ \Phi_2(\rho) &= (N - \kappa)\, M(-n_r, 2\gamma + 1, \rho) \end{aligned} \tag{2.48}$$

  where $M(a, b, \rho)$ is the confluent hypergeometric function which is defined by the series

$$\begin{aligned} M(a, b, \rho) &= 1 + \frac{a}{b}\rho + \frac{a(a+1)}{b(b+1)} \frac{\rho^2}{2!} + cdots \\ &= \sum_{m=0}^{\infty} \frac{(a)_m}{(b)_m} \frac{\rho^m}{m!} \end{aligned} \tag{2.49}$$

  and

$$\begin{aligned} (a)_m &= \frac{(a+m-1)!}{(a-1)!} \\ &= \frac{\Gamma(a+m)}{\Gamma(a)} \end{aligned} \tag{2.50}$$

- $c^2 E$ is the total energy. It is given by the Sommerfeld formula

$$\begin{aligned} E &= 1 - \frac{\epsilon}{c^2} \\ &= \frac{1}{\sqrt{1 + \left(\frac{\alpha Z}{n_r + \gamma}\right)^2}} \end{aligned} \tag{2.51}$$

- $n_r$ is known as the radial quantum number. It is a non-negative integer. This is necessary to ensure that the confluent hypergeometric functions are polynomials. The wavefunction will then correspond to bound-states. There is a discrete spectrum of energies with

$$\gamma_1 \leq E < 1 \tag{2.52}$$

where $\gamma_1 = \sqrt{1-(\alpha Z)^2}$ is the energy of the 1s orbital with quantum numbers $n_r = 0$ and $\kappa = -1$.

- $N$ is known as the 'apparent' principal quantum number. It is given by

$$\begin{aligned} N &= \frac{\alpha Z}{\sqrt{1-E^2}} \\ &= \frac{n_r+\gamma}{E} \\ &= \sqrt{(n_r+\gamma)^2+(\alpha Z)^2} \end{aligned} \tag{2.53}$$

Note that $N$ is not an integer. However in the nonrelativistic limit $E \to 1$ and $\gamma \to |\kappa|$ so that

$$N \to n_r + |\kappa| = n \tag{2.54}$$

where integer $n$ can be identified as the principal quantum number. Therefore

$$N = \sqrt{n^2 - 2\,n_r\,(|\kappa|-\gamma)} \leq n \tag{2.55}$$

- The outside factor $-S_\kappa$ (where $S_\kappa$ is the sign of $\kappa$) has been included to ensure that the large radial component of the Dirac wavefunction is positive when $r$ is small ( i.e. , $P' > 0$ as $r \to 0$). In the nonrelativistic limit this will agree with the normal convention for hydrogenic bound-states.

  Considering only the $\Phi$ functions, the sign of the Dirac wavefunction near the origin is determined by $N-\kappa-n_r$. When $\kappa < 0$ this becomes $N-n+2|\kappa|$ which is positive. However when $\kappa > 0$ it becomes $N-n$ which is negative.

**Screened** Coulomb functions are obtained by using $Z_e = Z - \sigma$ in place of the actual atomic number $Z$ of the system being considered. If different **screening numbers** $\sigma_a$ are used for different subshells then the condition eq.(2.13) will not be satisfied in general. In such cases the **Gram-Schmidt orthogonalisation procedure**

$$\begin{pmatrix} P_a \\ Q_a \end{pmatrix} \to \begin{pmatrix} P_a \\ Q_a \end{pmatrix} - \sum_{b<a} \delta_{\kappa_a \kappa_b}\, N(ab) \begin{pmatrix} P_b \\ Q_b \end{pmatrix} \tag{2.56}$$

followed by normalisation may be used to obtain an orthonormal set of basis functions.

2. In many cases orbitals calculated in a potential based upon the nonrelativistic Thomas-Fermi theory are better estimates than screened Coulomb functions. This is because

the Thomas-Fermi potential provides an estimate of the radial variation of screening of the nuclear field. The Thomas-Fermi potential is given by

$$Y_{\kappa_a} \rightarrow Y^{TF}(r) = Z_\infty - (rV_{nuc}(r) + Z_\infty)\,[f(x)]^2 \tag{2.57}$$

where

$$Z_\infty = Z + 1 - \sum_{a=1}^{n_o} q_{\rm av}(a) \tag{2.58}$$

$$f(x) = \frac{0.60112x^2 + 1.81061x + 1}{0.04793x^5 + 0.21465x^4 + 0.77112x^3 + 1.39515x^2 + 1.81061x + 1} \tag{2.59}$$

$$x = \left[\frac{(Z - Z_\infty)^{\frac{1}{3}}\, r}{0.8853}\right]^{\frac{1}{2}} \tag{2.60}$$

The **average occupation number** $q_{\rm av}(a)$ is defined by

$$q_{\rm av}(a) = \frac{\sum_{r=1}^{n_c} (2J_r + 1)\, q_r(a)}{\sum_{r=1}^{n_c} (2J_r + 1)} \tag{2.61}$$

3. Calculations based on density-functional theory may be considered to be the next level in sophistication, as they include an estimate of certain exchange and correlation effects. Now the potential $Y_{\kappa_a}(r)$ includes a term that is also a function of the spherically-averaged particle density $\rho(r)$

$$Y_{\kappa_a}(r) \rightarrow Y_{\kappa_a}(r) - Y^{\rm xc}_{\kappa_a}(\rho; r) \tag{2.62}$$

where

$$\rho(r) = \frac{1}{4\pi r^2} \sum_{a=1}^{n_o} q_{\rm av}(a)\, (P_a^2(r) + Q_a^2(r)) \tag{2.63}$$

The **Slater exchange approximation** consists in setting

$$Y^{\rm xc}_{\kappa_a}(\rho; r) \approx Y^{Slx}(\rho; r) = \tfrac{3}{2}\, \left(\tfrac{3}{\pi}\, \rho(r)\right)^{\frac{1}{3}} \tag{2.64}$$

In the last case the potential depends on the orbitals through $\rho$. This makes the differential equations nonlinear. Such systems are solved by a **self-consistent-field** (SCF) procedure such as:

**Step 1** The potential $Y_{\kappa_a}(r)$ is calculated from an estimated set of radial functions $\{P_b Q_b\}$

**Step 2** The differential equations are solved using this potential to obtain a new set of radial functions $\{P_b^{\rm est} Q_b^{\rm est}\}$

**Step 3** An improved estimated set of radial functions is obtained from

$$\begin{pmatrix} P_b^{\rm est} \\ Q_b^{\rm est} \end{pmatrix} \rightarrow (1-\eta_b) \begin{pmatrix} P_b^{\rm new} \\ Q_b^{\rm new} \end{pmatrix} + \eta_b \begin{pmatrix} P_b^{\rm est} \\ Q_b^{\rm est} \end{pmatrix} \tag{2.65}$$

where $0 \leq \eta_b < 1$ are **damping** or **accelerating** factors. If the improved estimated set agrees to within a specified tolerance with the original estimated set, **convergence** has been achieved. If not steps 1–3 are executed again.

When different orbitals with the same angular quantum number are generated in different potentials, we no longer have an eigenvalue problem: condition eq.(2.13) is enforced by introducing the inhomogeneous terms

$$X_a^P = \frac{r}{cq_{\rm av}(a)} \sum_{b \neq a} \delta_{\kappa_a \kappa_b}\, \epsilon_{ab}\, Q_b(r) \tag{2.66}$$

$$X_a^Q = \frac{r}{cq_{\rm av}(a)} \sum_{b \neq a} \delta_{\kappa_a \kappa_b}\, \epsilon_{ab}\, P_b(r) \tag{2.67}$$

and the required **Lagrange multipliers** $\epsilon_{ab}$ are determined from either of

$$\frac{\epsilon_{ab} r}{q_{\rm av}(a)} = \int_0^\infty \frac{dr}{r} \left(Y_a(r) + rV_{nuc}(r)\right) \left(P_a(r)P_b(r) + Q_a(r)Q_b(r)\right) - I(ab) \tag{2.68}$$

$$\frac{\epsilon_{ab} r}{q_{\rm av}(b)} = \int_0^\infty \frac{dr}{r} \left(Y_b(r) + rV_{nuc}(r)\right) \left(P_a(r)P_b(r) + Q_a(r)Q_b(r)\right) - I(ab) \tag{2.69}$$

or their difference or sum.

Koopmans has shown [13], quite generally, that Lagrange multipliers need be included only between pairs of orbitals $(a, b)$ that vary subject to eq.(2.13) if either $\bar{q}(a) < 2j_a + 1$ or $\bar{q}(b) < 2j_b + 1$. For pairs involving one fixed orbital Lagrange multipliers must always be included. Pairs in which both orbitals are fixed are assumed orthogonal.

It is now necessary to calculate the inhomogeneous terms also in Step 1 of the SCF procedure given above.

The most general form of the differential equations arises when such equations are derived from a variational principle. Consider the energy functional

$$W^{DC} = \sum_{r,s=1}^{n_c} w_{rs}\, H_{rs}^{DC} + \sum_{a=1}^{n_o} \bar{q}(a)\, \epsilon_a\, N(aa) + \sum_{a=1}^{n_o} \sum_{b=a+1}^{n_o} \delta\kappa_a\kappa_b\, \epsilon_{ab}\, N(ab) \tag{2.70}$$

with **generalised weights**

$$w_{rs} = \frac{\sum_{i=1}^{n_L} (2J_i + 1)\, c_{r\Gamma_i}\, c_{s\Gamma_i}}{\sum_{i=1}^{n_L} (2J_i + 1)} \tag{2.71}$$

This is simply equivalent to

$$W^{DC} = \frac{\sum_{i=1}^{n_L} (2J_i + 1)\ E^{DC}_{\Gamma_i}}{\sum_{i=1}^{n_L} (2J_i + 1)} \tag{2.72}$$

a weighted sum over a certain subset of atomic levels with Lagrange multipliers $\epsilon_a$ and $\epsilon_{ab}$ introduced to enforce the restriction (6). The **generalised occupation numbers** $\bar{q}(a)$ are defined in terms of the diagonal coefficients

$$\bar{q}(a) = \sum_{r=1}^{n_c} w_{rr}\ q_r(a) \tag{2.73}$$

The requirement that $W^{DC}$ be stationary with respect to variations in the radial functions $\{P_a Q_a\}$ leads to equations of the form (28) with the **direct potential** $Y_a(r)$ given by

$$Y_a(r) = -rV_{nuc}(r) - \sum_k \left( \sum_{b=1}^{n_o} y^k(ab)\ Y^k(ab;r) - \sum_{b,d=1}^{n_o} y^k(abad)\ Y^k(bd;r) \right) \tag{2.74}$$

$$y^k(ab) = \frac{1+\delta_{ab}}{\bar{q}(a)} \sum_{r=1}^{n_c} w_{rr}\ f^k_r(ab) \tag{2.75}$$

$$y^k(abad) = \frac{1}{\bar{q}(a)} \sum_{r,s=1}^{n_c} w_{rs}\ V^k_{rs}(abad) \tag{2.76}$$

and the inhomogeneous terms

## 2.7 Breit interaction

General open-shell matrix elements take the form

$$\begin{aligned}
\langle TM|H|T'M'\rangle &= \delta_{JJ'}\delta_{MM'} \sum_{abcd} (-)^{\Delta} \sqrt{N_a(N_b - \delta_{ab})N_c(N_d - \delta_{cd})} \\
&\times \sum_{\bar{T}} (T_a\{|\bar{T}_a j_a)\,(T_b\{|\bar{T}_b j_b)\,(\bar{T}_c j_c|\}T'_c)\,(\bar{T}_d j_d|\}T'_d) \\
&\times \sum_k \Big\{ C_d\,(1+\delta_{ab}\delta_{cd})^{-1}\,[j_a, j_d]^{-\frac{1}{2}}\,X^k(abcd) \\
&\times -C_e\,(1-\delta_{ab})(1-\delta_{cd})\,[j_a, j_c]^{-\frac{1}{2}}\,X^k(abdc) \Big\}
\end{aligned}$$

For the transverse Breit interaction

$$\begin{aligned}
X^k(abcd) &= (-)^k\,\langle j_a||\boldsymbol{C}^{(k)}||j_c\rangle\,\langle j_b||\boldsymbol{C}^{(k)}||j_d\rangle \\
&\times \Delta(j_a j_c k)\,\Delta(j_b j_d k)
\end{aligned}$$

$$\times \quad \left\{ \sum_{\nu=k-1}^{k+1} \Pi(\kappa_a\kappa_c\nu)\,\Pi(\kappa_b\kappa_d\nu) \sum_{\mu=1}^{4} r_\mu^{\nu k}(abcd)\,R_\mu^\nu(abcd) \right.$$
$$\left. +\Pi(\kappa_a\kappa_c, k-1)\,\Pi(\kappa_b\kappa_d, k+1) \sum_{\mu=1}^{8} s_\mu^k(abcd)\,S_\mu^k(abcd) \right\}$$

Now define the following radial integrals

$$R^k[ac|bd] = \int_0^\infty \int_0^\infty \rho_{ac}(r)\,\tfrac{1}{2}\{V_k(r,s;\omega_{ac}) + V_k(r,s;\omega_{bd})\}\,\rho_{bd}(s)\,dr\,ds$$

and

$$S^k[ac|bd] = \int_0^\infty \int_0^\infty \rho_{ac}(r)\,\tfrac{1}{2}\{W_k(r,s;\omega_{ac}) + W_k(r,s;\omega_{bd})\}\,\rho_{bd}(s)\,dr\,ds$$

where

$$\begin{aligned}
\rho_{ac}(r) &= P_a(r)\,Q_c(r) \\
V_k(r,s;\omega) &= -(2k+1)\,\omega\,j_k(wr_<)\,n_k(wr_>) \\
W_k(r,s;\omega) &= \begin{cases} (2k+1)\,\omega\,j_{k-1}(wr)\,n_{k+1}(ws) + (\frac{2k+1}{\omega})^2\,\frac{r^{k-1}}{s^{k+2}} & \text{when } r<s \\ (2k+1)\,\omega\,j_{k+1}(ws)\,n_{k-1}(wr) & \text{when } r>s \end{cases}
\end{aligned}$$

The summation label $\mu$ refers to permutations of the orbital labels

$$\begin{aligned}
R_\mu^k(abcd) = \quad & R^k[ac|bd] \quad \text{when } \mu = 1 \\
& R^k[ca|db] \quad \text{when } \mu = 2 \\
& R^k[ac|db] \quad \text{when } \mu = 3 \\
& R^k[ca|bd] \quad \text{when } \mu = 4
\end{aligned}$$

and

$$\begin{aligned}
S_\mu^k(abcd) = \quad & S^k[ac|bd] \quad \text{when } \mu = 1 \\
& S^k[bd|ac] \quad \text{when } \mu = 2 \\
& S^k[ca|db] \quad \text{when } \mu = 3 \\
& S^k[db|ca] \quad \text{when } \mu = 4 \\
& S^k[ac|db] \quad \text{when } \mu = 5 \\
& S^k[db|ac] \quad \text{when } \mu = 6 \\
& S^k[ca|bd] \quad \text{when } \mu = 7 \\
& S^k[bd|ca] \quad \text{when } \mu = 8
\end{aligned}$$

The coefficients $r_\mu^{\nu k}(abcd)$ are given by

| | $\nu = k-1$ | $k$ | $k+1$ |
|---|---|---|---|
| $\mu = 1$ | $A(K+k)(K'+k)$ | $A$ | $A(K-k-1)(K'-k-1)$ |
| 2 | $A(K-k)(K'-k)$ | $A$ | $A(K+k+1)(K'+k+1)$ |
| 3 | $A(K+k)(K'-k)$ | $A$ | $A(K-k-1)(K'+k+1)$ |
| 4 | $A(K-k)(K'+k)$ | $A$ | $A(K+k+1)(K'-k-1)$ |

The coefficients $s^k_\mu(abcd)$ are given by

| | |
|---|---|
| $\mu = 1$ | $B(K+k)(K'-k-1)$ |
| 2 | $B(K'+k)(K-k-1)$ |
| 3 | $B(K-k)(K'+k+1)$ |
| 4 | $B(K'-k)(K+k+1)$ |
| 5 | $B(K+k)(K'+k+1)$ |
| 6 | $B(K'-k)(K-k-1)$ |
| 7 | $B(K-k)(K'-k-1)$ |
| 8 | $B(K'+k)(K+k+1)$ |

In these formulae

$$
\begin{aligned}
K &= \kappa_c - \kappa_a \\
K' &= \kappa_d - \kappa_b \\
A &= \begin{cases} (k+1)/k(2k-1)(2k+1) & \text{when } \nu = k-1 \\ -(\kappa_a+\kappa_c)(\kappa_b+\kappa_d)/k(k+1) & \text{when } \nu = k \\ k/(k+1)(2k+1)(2k+3) & \text{when } \nu = k+1 \end{cases} \\
B &= 1/(2k+1)^2
\end{aligned}
$$

# Chapter 3

# GRASP$^2$ input

## 3.1 GRASP$^2$ input data summary

•: Compulsory at its level
◦: Optional at its level

1. • TIME Card
2. ◦ Problem Specification Card Group:
   (a) • Title Card
   (b) • CSF Manifold Specification and Print Control Card Group:
      i. • Convention/Dimension/Transformation/Print Control Card
      ii. • Configuration Specification Card(s)
      iii. • Coupling Specification and Angular Module Print Control Card Group:
         A. • ANG Card
         B. • Coupling Information Card(s)
   (c) ◦ MCP Invocation and Unit Specification Card
   (d) ◦ MCDF Card Group:
      i. • MCDF Invocation and Option Specification Card
      ii. • MCDF Unit Specification Card
      iii. • Nuclear Charge/Mass Card
      iv. ◦ NUCLEUS Card
      v. ◦ FIX Card
      vi. ◦ LOAD Card
      vii. ◦ ORBOUT Card
      viii. ◦ SCF Card
      ix. ◦ GRID Card
      x. ◦ RLDA Card
      xi. ◦ PRINT Card (absent here if used in (f) iii.)
      xii. ◦ SCREEN Card
      xiii. ◦ METHOD Card
      xiv. ◦ CDAMP Card
      xv. ◦ ODAMP Card
      xvi. ◦ CPOLPOTL Card Group:
         A. • CPOLPOTL Invocation Card
         B. • Core-Polarisation Potential Data Cards
      xvii. ◦ NOINVERT Card
      xviii. ◦ THRESHLD Card
      xix. • Calculation Type Card
   (e) ◦ MCBP Invocation and Unit Specification Card

(f) ∘ BENA Card Group:

i. • BENA Invocation and Option Specification Card
ii. • BENA Unit Specification Card
iii. ∘ PRINT Card (absent here if used in (d) xi.)
iv. ∘ LOW1 Card

(g) ∘ MCT Invocation and Unit Specification Card

(h) ∘ OSCL Card Group:

i. • OSCL Invocation and Option Specification Card
ii. • OSCL Unit Specification Card
iii. ∘ LEV Card
iv. ∘ CUT Card
v. ∘ LOW2 Card

(i) • END Card

---

## 3.2 Card structure summary

### 3.2.1 Notation

*c*: Character string
*i*: Integer constant
*n*: Numerical constant, of type *i*, or floating or fixed-point
*s*: Orbital specification: form is *ic* with no space between the *i* and the *c*
*i* is the principal quantum number and must be a positive integer, and *c* must be one of the following *symmetries*:

| Symmetry | S | P- | P | D- | D | F- | F | G- | G | H- | H |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *l* | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 |
| *j* | 1/2 | 1/2 | 3/2 | 3/2 | 5/2 | 5/2 | 7/2 | 7/2 | 9/2 | 9/2 | 11/2 |
| Parity | *e* | *o* | *o* | *e* | *e* | *o* | *o* | *e* | *e* | *o* | *o* |

| Symmetry | I- | I | K- | K | L- | L | M- | M | N- | N |
|---|---|---|---|---|---|---|---|---|---|---|
| *l* | 6 | 6 | 7 | 7 | 8 | 8 | 9 | 9 | 10 | 10 |
| *j* | 11/2 | 13/2 | 13/2 | 15/2 | 15/2 | 17/2 | 17/2 | 19/2 | 19/2 | 21/2 |
| Parity | *e* | *e* | *o* | *o* | *e* | *e* | *o* | *o* | *e* | *e* |

Symmetries involving the - sign cannot be used to specify nonrelativistic (*nl*) orbitals.

### 3.2.2 Facilities in CARDIN

$$
\begin{aligned}
i{*}n &\equiv \underbrace{n \ldots n}_{i \text{ times}} \\
i{*}(\ldots) &\equiv \underbrace{(\ldots) \ldots (\ldots)}_{i \text{ times}} \\
n_1/n_2 &\equiv n_1 \div n_2 \\
n_1{-}n_2 &\equiv \begin{cases} n_1, n_1+1, \ldots, n_2 \text{ if } n_1 < n_2 \\ n_1, n_1-1, \ldots, n_2 \text{ if } n_1 > n_2 \end{cases}
\end{aligned}
$$

The - *must* be isolated by spaces when used to specify a range. An isolated `C` implies that a continuation line follows; the card may be spread over any number of lines.

---

### 3.2.3 TIME Card

Format: `TIME` $n$

$n$**:** CPU seconds available for run.

---

### 3.2.4 Title Card

Format: $c$

$c$**:** 0–72-character string. This string is written as an identifier in all GRASP$^2$ files.

---

### 3.2.5 Convention/Dimension/Transformation/Print Control Card

Format 1: {2 or 3}$i$

$i_1$**:** Number of CSF manifolds ($n_\mathrm{c}$).

$i_2$**:** Number of orbitals ($n_\mathrm{w}$).

$i_3$**:** Convention/transformation/print-control index:

`0`: (Default) 'relativistic' orbitals (maximum occupation $(2j+1)$).

`1`: 'Nonrelativistic' orbitals (maximum occupation $2(2l+1)$).

`2`: As for 1; CSF manifolds are defined in a new coupling scheme, the transformation matrix between the default and the new CSF manifold is calculated and printed, and the eigenvectors are printed in the new basis

`3`: As for 2, but the transformation matrix is not printed

or

Format 2: $\{3\}i$ $c$'s

$i_1$: Number of configurations ($n_\mathrm{c}$).

$i_2$: Number of orbitals ($n_\mathrm{w}$).

$i_3$: Unit number for file to dump transformation information and matrix. Set to `0` if not used.

$c$:

`NOHEAD`: Do not print CSF manifold headings.

`HEAD`: Print CSF manifold headings (default).

`OLDONLY`: Print *standard* $jj$-coupled CSF manifold data and not those in other basis.

`NEWONLY`: Print CSF manifold data only in new basis.

`CSFONLY`: Print only the CSF manifold data in both schemes: do not print the transformation coefficients (default).

`NOCOEFF`: Do not print transformation coefficients.

`COEFF`: Print transformation coefficients.

`ALL`: Print CSF manifold data in both bases and transformation coefficients.

`NOPRINT`: Do not print anything.

`NOFILE`: Do not dump to unit $i_3$ (default if $i_3 = 0$).

`FILE`: Dump to unit $i_3$ (default if $i_3 \neq 0$).

---

### 3.2.6 Configuration Specification Card

There should be $n_\mathrm{w}$ such cards.

1. Format: $s$

   Orbital $s$ is full in all CSF manifolds.

2. Format: $s$ $i$

   Orbital $s$ has occupation $i$ in all CSF manifolds.

3. Format: $s$ $\{n_c\}i$

   $i_k$: Occupation number of orbital $s$ in CSF manifold $k$.

---

### 3.2.7 ANG Card

Format: `ANG` $\{0\text{–}7\}i$

**1** : Print out all angular coefficients calculated: $T_{rs}(ab)$, $V^k_{rs}(abcd)$ in MCP, $V^{k\tau}_{rs}(abcd)$ in MCBP, $d^{\pi k}_{rs}(ab)$ in MCT.

**2** : Debug output from routines RKCO, etc. .

**3** : Debug output from routines in NJGRAF.

**4** : Debug output from VIJOUT.

**5** : Debug output from TMSOUT.

**6** : Debug output from TNSRJJ.

**7** : Suppress call to CFOUT.

---

### 3.2.8 Coupling Information Card

Format 1: $i$ $n$'s

This format *must* be used if Format 1 is selected for the Convention/Dimension/Transformation/Print Control Card.

$i$**:** Sequence number (going from left to right on the Configuration Specification Card(s)) of the configuration for which coupling information is being provided.

$n$**:** $J$ (and $v$ when required) and $X$ sequences required to fully specify the standard coupling scheme. If there are no open subshells, these *cards* must be absent. If there is a single open subshell, only its $J$ value must be specified. If there is more than one open subshell the appropriate set of $J$'s and $X$'s must be specified.

The following tables help in deciding the entries for format 1 of the coupling information cards.

A *subshell* is defined by the angular momentum, $j$, of the orbitals that constitute it, and its occupation number, $0 \le q \le 2j+1$. The $q$ orbitals are $jj$-coupled to give total angular momentum quantum numbers $J$, $M$, and seniority $v$:

$$|(j)^q\, v\, J\, M\rangle \tag{3.1}$$

GRASP[2] is limited (by data set in `BLOCK DATA TERMS`) to $\frac{1}{2} \le j \le \frac{15}{2}$, and further, $2 \le q \le 2j-1$ for $j \ge \frac{9}{2}$. If using Format 1 for the coupling information cards, seniority, $v$, need be specified *if and only if* $j = \frac{7}{2}$, $q = 4$, $J = 2, 4$.

| $j$ | $q$ | $v$ | $J$ |
|---|---|---|---|
| $\frac{1}{2}$ | 0, 2 | 0 | 0 |
| | 1 | 1 | $\frac{1}{2}$ |
| $\frac{3}{2}$ | 0, 4 | 0 | 0 |
| | 1, 3 | 1 | $\frac{3}{2}$ |
| | 2 | 0 | 0 |
| | | 2 | 2 |
| $\frac{5}{2}$ | 0, 6 | 0 | 0 |
| | 1, 5 | 1 | $\frac{5}{2}$ |
| | 2, 4 | 0 | 0 |
| | | 2 | 2, 4 |
| | 3 | 1 | $\frac{5}{2}$ |
| | | 3 | $\frac{3}{2}$, $\frac{9}{2}$ |
| $\frac{7}{2}$ | 0, 8 | 0 | 0 |
| | 1, 7 | 1 | $\frac{7}{2}$ |
| | 2, 6 | 0 | 0 |
| | | 2 | 2, 4, 6 |
| | 3, 5 | 1 | $\frac{7}{2}$ |
| | | 3 | $\frac{3}{2}$, $\frac{5}{2}$, $\frac{9}{2}$, $\frac{11}{2}$, $\frac{15}{2}$ |
| | 4 | 0 | 0 |
| | | 2 | 2, 4, 6 |
| | | 4 | 2, 4, 5, 8 |

| $j$ | $q$ | $v$ | $J$ |
|---|---|---|---|
| $\frac{9}{2}$ | 0, 10 | 0 | 0 |
| | 1, 9 | 1 | $\frac{9}{2}$ |
| | 2, 8 | 0 | 0 |
| | | 2 | 2, 4, 6, 8 |
| $\frac{11}{2}$ | 0, 12 | 0 | 0 |
| | 1, 11 | 1 | $\frac{11}{2}$ |
| | 2, 10 | 0 | 0 |
| | | 2 | 2, 4, 6, 8, 10 |
| $\frac{13}{2}$ | 0, 14 | 0 | 0 |
| | 1, 13 | 1 | $\frac{13}{2}$ |
| | 2, 12 | 0 | 0 |
| | | 2 | 2, 4, 6, 8, 10, 12 |
| $\frac{15}{2}$ | 0, 16 | 0 | 0 |
| | 1, 15 | 1 | $\frac{15}{2}$ |
| | 2, 14 | 0 | 0 |
| | | 2 | 2, 4, 6, 8, 10, 12, 14 |

or

Format 2: $n$ {0–2}$i$ $\underbrace{l \text{ or } c}_{\text{optional}}$ {0–$n_{\text{open}}$}{$i$ or $s$}

This format *must* be used if Format 2 is selected for the Convention/Dimension/Transformation/Print Control Card.

There must be $n_c$ Coupling Information Cards.

$n$**:** Total angular momentum quantum number, $J$ (integer or half-integer), for configuration; if this is a negative number CSF manifolds with all possible $J$ will be included.

$i_1$**:** Number of open subshells (integer) to be transformed from $LS$- to $LSJ$-coupling; If this is less than $n_{\text{open}}$, the outermost $i_2$ subshells are chosen.

$i_2$**:** Number of open subshells (integer) to be transformed from $LSJ$- to $LLSS$-coupling; this must be $\leq i_1$; if $< i_1$, the outermost $i_2$ subshells are chosen.

$l$ or $c$**:** `TRUE`, `T`, `FALSE`, `F`, or `J-L` to indicate whether $jl$-coupling is to be used on the outermost open subshell.

{0–$n_{\text{open}}$}$i$**:** List of subshells in order of coupling. Subshells may be defined by the open subshell index number, the orbital label, or the subshell index number. Parentheses may be used to indicate that a pair of subshells is to be coupled together before being coupled to the previous resultant angular momentum. The list of subshells should always be given with the minimum number of parentheses. The proper order of the subshells is always maintained when the coupling tree is set up: subshells with lower index are always placed to the left of those with higher index numbers.

### 3.2.9 MCP Invocation and Unit Specification Card

Format: `MCP` {2–3}$i$

$i_1$**:** Unit number of the UNSORTED MCP OUTPUT FILE. This stores the CSF-manifold-based list of $T_{rs}(ab)$ and $V^k_{rs}(abcd)$ coefficients.

$i_2$**:** Unit number of the SORTED MCP OUTPUT FILE. This stores the integral-based list of $T_{rs}(ab)$ and $V^k_{rs}(abcd)$ coefficients.

$i_3$**:** Unit number of the UNSORTED MCP INPUT FILE. This is an UNSORTED MCP OUTPUT FILE from an incomplete run.

If $i_1 = i_3$ this is the UNSORTED MCP INPUT/OUTPUT FILE.

---

### 3.2.10 MCDF Invocation and Option Specification Card

Format: `MCDF` {0–33}$i$

i:

**1** : Print the radial grid and associated arrays.

**2** : Print $V_{\rm nuc}(r)$.

**3** : Print $V^{\rm TF}(r)$.

**4** : Print the orbital wavefunctions at the end of each SCF iteration.†

**5** : Print the coefficients of the direct and exchange potentials each time they are calculated.

**6** : Print the direct and exchange potentials each time they are calculated.†

**7** : Print iteration information in routine SOLVE.

**8** : Print orbital wavefunctions on each iteration in routine SOLVE.†

**9** : Print $\boldsymbol{H}^{\rm DC}$ each time it is calculated.†

**10** : Print out $E^{\rm DC}_\Gamma$ and $\boldsymbol{c}^{\rm DC}_\Gamma$ each time they are calculated.†

**11** : Print the $R^k(abcd)$ each time they are calculated.†

**12** : Print the $I(ab)$ each time they are calculated.†

**13** : Print the $Y^k(ab;r)$ each time they are calculated.†

**14** : Print nonzero coefficients in routines FCO and GCO.

**15** : Print the composition of the direct and exchange potentials.†

**16** : Print a performance message in routine INTRPQ.

**17** : Print a performance message in routine DPBDT (Aitken version only).

**18** : Print a performance message in routine QUAD.

**19** : Print a performance message in routine TAIL.

**22** : Append READ ORBITALS file to WRITE ORBITALS file before writing out orbitals.

**24** : Print orbitals at convergence.

**25** : Enforce weak orthogonality (Schmidt orthogonalise orbitals only at the end of an SCF cycle).

**26** : Omit all Lagrange multipliers.

**27** : Perform rotation analysis.

**28** : Enforce strong orthogonality (Schmidt orthogonalise an orbital as soon as it is calculated).

**29** : Compute isotope-shift parameters $\partial E_\Gamma/\partial c$ and $\partial E_\Gamma/\partial a$.

**31** : Suppress the printout of $jj$-coupled eigenvectors.

**32** : Suppress the printout of weights in the $jj$-coupled basis.

**33** : Suppress the printout of the recoupled eigenvectors.

**34** : Suppress the printout of weights in the recoupled basis.

**35** : Suppress the printout of nonrelativistic configuration weights.

**36** : Suppress printout of interlevel separations.

**37** : Suppress printout of level energies relative to lowest level.

† *Warning*: This can produce a very large printout.

---

### 3.2.11 MCDF Unit Specification Card

Format: $\{3\text{–}5\}i$

$i_1$**:** Unit number of the SORTED MCP (OUTPUT) FILE. See the MCP Invocation and Unit Specification Card.

$i_2$**:** Unit number of the MCDF READ FILE; set to `0` if this does not exist. This is an MCDF DUMP FILE from an incomplete run.

$i_3$**:** Unit number of the MCDF DUMP FILE. This file stores all information required to restart the SCF iterations if convergence has not been attained.

$i_4$**:** Unit number of the ORBOUT READ FILE; this *must* be nonzero if a `1` appears on the LOAD Card.

$i_5$**:** Unit number of the ORBOUT DUMP FILE; this *must* be nonzero if the ORBOUT Card is used. Orbital parameters, orbitals, and the radial grid are written to the ORBOUT DUMP FILE at convergence.

If $i_2 = i_3$ this is the MCDF READ/DUMP FILE.

If $i_4 = i_5$ this is the ORBOUT READ/DUMP FILE.

---

### 3.2.12 Nuclear Charge/Mass Card

Format: $\{1 \text{ or } 2\}n$

$n_1$**:** Atomic number $Z$.

$n_2$**:** Nuclear mass $M_{\text{nuc}}$ in amu; if nonzero, the default nucleus type is `FERMI`; if omitted, $M_{\text{nuc}} = \infty$ and the nucleus type is `POINT`.

---

### 3.2.13 NUCLEUS Card

Format 1: `NUCLEUS POINT`

or

Format 2: `NUCLEUS FERMI` $\{0 - 2\}n$

$n_1$: Root mean square radius, $\langle r^2 \rangle^{1/2}$, of the Fermi nuclear charge distribution in fm; the default is $0.836\, M_{\rm nuc}^{1/3} + 0.570$. The mean square radius is related to the parameters $c$ and $a$ through $\langle r^2 \rangle = (3/5)\, c^2 + (7/5)\, \pi^2 a^2$.

$n_2$: Skin thickness, $t$, of the Fermi charge distribution in fm; the default is 2.30 fm. The skin thickness is related to the parameter $a$ through $t = a \cdot 4 \ln 3$.

See the Nuclear Charge/Mass Card for the default nucleus type.

---

### 3.2.14 FIX Card

Format: `FIX` $\{1$–$n_{\rm w} - 1\} i$

$i$: Sequence number of orbital that must be held fixed (frozen). This card should *not* be used if *all* orbitals are to be held fixed; use the `CI` Calculation Type Card instead.

The default is to hold no orbitals fixed.

---

### 3.2.15 LOAD Card

Format: `LOAD` $\{n_{\rm w}\} i$

$i_a$:

`0`: Read orbital $a$ from the MCDF READ (or READ/DUMP) FILE.

`1`: Read orbital $a$ from the ORBOUT READ (or READ/DUMP) FILE.

`2`: Generate orbital $a$ using a nonrelativistic Thomas-Fermi potential; this is the default if the LOAD Card is not used.

`3`: use a screened Coulomb function for orbital $a$.

Screening parameters can be set with the SCREEN Card if the default (Fischer) estimation procedure is not chosen.

---

### 3.2.16 ORBOUT Card

Format: `ORBOUT` $\{0$–$n_{\mathrm{w}}\}i$

$i$**:** The sequence number of an orbital to be appended to the ORBOUT DUMP (or READ/DUMP) FILE. See the MCDF Invocation and Option Specification Card. If omitted, *all* orbitals are appended to this file. Orbitals that are held fixed (see the FIX Card) cannot be written out. Option 22 on the MCDF Invocation and Option Specification Card affects the structure of the ORBOUT DUMP (or READ/DUMP) FILE.

The default is to append no orbitals to the ORBOUT DUMP (or READ/DUMP) FILE.

---

### 3.2.17 SCF Card

Format: `SCF` $\{1$–$5\}\{c\,n\}$

$c$**:** `ACCY`, `CON`, `NSCF`, `NSIC`, or `NSOLV`.

$n$**:** Accuracy criterion ($> 0$) if this follows `ACCY`, default is $h^6$; factor by which the speed of light is to be multiplied ($> 0$) if this follows `CON`: the nonrelativistic limit can usually be obtained with this set to $10^3$; number of SCF iterations ($\geq 1$) if this follows `NSCF`; number of improvements after the sweep through all functions in each SCF iteration if this follows `NSIC`; maximum number of iterations in routine SOLVE if this follows `NSOLV`: default is thrice the principal quantum number for a given orbital.

---

### 3.2.18 GRID Card

Format: `GRID` $\{1$–$4\}\{c\,n\}$

The radial grid has the form:

$$r_k = r_0(e^{(k-1)h} - 1) \qquad \text{where} \qquad k = 1, \ldots, n_p \tag{3.2}$$

if $h'$ is 0, and the form

$$\ln(r_k/r_0 + 1) + (h/h')r_k = (k-1)h \qquad \text{where} \qquad k = 1, \ldots, n_p \tag{3.3}$$

otherwise.

$c$**:** `RNT`, `H`, `HP`, or `N`.

$n$: If preceded by `RNT`, this is the grid parameter $r_0(> 0)$; if preceded by `H` this is the grid parameter $h(> 0)$; if preceded by `HP`, this is the grid parameter $h'(> 0)$; if preceded by `N` this is the grid parameter $n_p$.

If a Fermi nucleus is used, the defaults are $r_0 = 2 \times 10^{-6}$, $h = 0.05$, $n_p =$ `NP`; the defaults for the point nucleus case are $r_0 = e^{-65/16}/Z$, $h = 0.0625$, $n_p = 220$, if $h' = 0$, and $n_p =$ `NP` otherwise. The default for $h'$ in either case is 0.

---

### 3.2.19 RLDA Card

Format: `RLDA` $\{i\ n\}$

$i$:

- `0` for nonrelativistic statistical exchange only;
- `1` for relativistic statistical exchange only;
- `2` (default) for relativistic statistical exchange and correlation.

$n$: Exchange parameter: e.g., `1` (default) for Kohn-Sham exchange; `3/2` for Slater exchange.

The default is to use the usual direct and exchange potentials.

---

### 3.2.20 PRINT Card

Format: `PRINT` $\{0\text{–}n_\mathrm{c}\}i$

$i_k$: Print information related to this level. The levels are ordered by increasing energy.

If no levels are specified, information is printed for all $n_\mathrm{c}$ levels.

The default is printing for all $n_\mathrm{c}$ levels for `AL` or `EAL` calculations, $i$ for `OL` and $i(j)$ with $j = 1, \ldots, n_\mathrm{L}$ for `EOL` calculations.

---

### 3.2.21 SCREEN Card

Format 1: `SCREEN` $\{1\text{–}n_{\mathrm{w}}\}\{i\,n\}$

or

Format 2: `SCREEN` $i$

$i$**:** Sequence number of an orbital. This is absent in Format 2, where $n$ applies to *all* orbitals.

$n$**:** Screening number $\sigma_i$. Fischer has defined the default procedure for calculating this quantity.

---

### 3.2.22 METHOD Card

Format 1: `METHOD` $\{1\text{–}n_{\mathrm{w}}\}\{i_1\,i_2\}$

or

Format 2: `METHOD` $i_2$

$i_1$**:** Orbital sequence number. This is absent in Format 2, where $i_2$ applies to *all* orbitals.

$i_2$**:** `1`, `2`, `3` or `4`: methods of solving the differential equations as specified by Fischer (1986).

Method `1` is the default method. This is suitable for cases where the Dirac equations are inhomogeneous and where self-consistency is $\approx 10^{-4}$ or worse. Homogeneous equations can be solved only with method `2`, and the program will detect this condition automatically. In many cases, once self-consistency is better than $\approx 10^{-4}$ this method will lead to the quickest convergence.

Method `3` is the same as method `1` except that node counting is *not* enforced. The user must therefore determine from the output listing (under the heading `NNP`) that the correct number of nodes, $n - l - 1$, has been attained.

Method `4` is the same as method `2` except that node-counting is again not enforced.

---

### 3.2.23 CDAMP Card

Format 1: `CDAMP` $\{1\text{–}n_{\mathrm{L}}\}\{i\,n\}$

or

Format 2: `CDAMP` $n$

*i*: Level sequence number in order of increasing energy. This is absent in Format 2, where $n$ applies to *all* levels.

*n*: Damping factor, $\xi_i$ ($\texttt{0} < \xi_i < \texttt{1}$), for level $i$.

This card has no effect on `AL` or `EAL` calculations. For `OL` and `EOL` calculations, the default is $\xi_i = \texttt{0}$ for all $j = 1, \ldots, n_{\mathrm{L}}$ levels $i(j)$.

---

### 3.2.24 ODAMP Card

Format 1: `ODAMP` $\{1\text{–}n_{\mathrm{w}}\}\{i\, n\}$

or

Format 2: `ODAMP` $n$

*i*: Orbital sequence number as defined by the Configuration Specification Cards. This is absent in Format 2, where $n$ applies to *all* orbitals.

*n*: The damping factor, $\eta$, ($\texttt{-1} < \eta < \texttt{1}$, but $\eta \neq \texttt{0}$) for this orbital is $|n|$. The negative sign indicates a constant damping factor. Otherwise the program adjusts the damping as the calculation progresses. The default is $\eta = \texttt{0}$.

---

### 3.2.25 CPOLPOTL Invocation Card

Format: `CPOLPOTL` $i$

*i*: Number of orbitals that suffer a core-polarisation potential. The default is `0`.

---

### 3.2.26 Core-Polarisation Potential Data Card

Format: $s$ $\{2 \text{ or } 4\}n$

$n_1$: Dipole polarisability, $\alpha_{\mathrm{d}}$, of core of orbital $s$.

$n_2$: Cutoff radius, $r_{\mathrm{c}}$, of polarisation potential for this orbital.

$n_3$**:** Dynamical parameter, $\beta$, for this orbital. The default value is `0`.

$n_4$**:** Quadrupole polarisability, $\alpha_{\rm q}$, of this orbital's core. The default value is `0`.

---

### 3.2.27 NOINVERT Card

Format: `NOINVERT` $\{0 - n_{\rm w}\}i$

$i$**:** Sequence number of orbital not subject to the positive-slope-at-the-origin condition if method 3 or 4 has been selected. If omitted, all orbitals are selected. This card should not be used if methods 3 or 4 are not employed.

The default is that all orbitals are subject to the positive-slope-at-the-origin condition.

---

### 3.2.28 THRESHLD Card

Format: `THRESHLD` $n$

$n$**:** Oscillations in a large-component function, $P_{n_a\kappa_a}$, will be ignored if they have amplitudes less than $n$ times the maximum amplitude of this function. The default value of $n$ is 0.05.

---

### 3.2.29 Calculation Type Card

1. Format: `AL`

   Average Level calculation: MCDF procedure with

   $$d_{rs} = \delta_{rs}\,\frac{w_r}{\sum_{t=1}^{n_{\rm c}} w_t} \tag{3.4}$$

   with $w_t = 2J_t + 1$.

2. Format: `CI`

   Configuration Interaction calculation: no MCDF procedure.

3. Format: `OL` $i$

   Optimal Level calculation: MCDF procedure. The atomic levels are arranged in order of increasing energy after each diagonalisation of $\boldsymbol{H}^{\mathrm{DC}}$. With $n_{\mathrm{L}} = 1$, only level $i$ in this sequence contributes to the variational functional:

$$d_{rs} = c_{r\Gamma_i}\, c_{s\Gamma_i} \tag{3.5}$$

4. `EAL`

   Extended Average Level calculation: MCDF procedure.

   (a) Format 1: `EAL 1`

$$d_{rs} = \delta_{rs}/n_{\mathrm{c}} \tag{3.6}$$

   (b) Format 2: `EAL` $\{n_{\mathrm{c}}\}n$

$$d_{rs} = \delta_{rs}\, \frac{n_r}{\sum_{t=1}^{n_{\mathrm{c}}} n_t} \tag{3.7}$$

5. `EOL`

   (a) Format 1: `EOL` $i_1$ $\{i_1\}i$

   (b) Format 2: `EOL` $i_1$ $\{i_1\}i$ `1`

   (c) Format 3: `EOL` $i_1$ $\{i_1\}\{i\, n\}$

   Extended Optimal Level calculation: MCDF procedure. The atomic levels are arranged in order of increasing energy after each diagonalisation of $\boldsymbol{H}^{\mathrm{DC}}$. Only $n_{\mathrm{L}} < n_{\mathrm{c}}$ levels $i(j)$, $j = 1, \ldots, n_{\mathrm{L}}$ in this sequence contribute to the variational functional:

$$d_{rs} = \frac{\sum_{j=1}^{n_{\mathrm{L}}} w_{i(j)}\; c_{r\Gamma_{i(j)}}\; c_{s\Gamma_{i(j)}}}{\sum_{j=1}^{n_{\mathrm{L}}} w_{i(j)}} \tag{3.8}$$

   $i_1$: Number of levels, $n_{\mathrm{L}}$, to be optimised upon.
   $i$: Level contributing to the variational functional.
   $n$: Weight $w_i$ of level $i$. If Format 1 is used, the weight of level $i$ is $w_i = 2J_i + 1$; if Format 2 is used, $w_i = 1$.

---

### 3.2.30 MCBP Invocation and Unit Specification Card

Format: `MCBP` {1 or 2}$i$

$i_1$**:** Unit number of the MCBP OUTPUT FILE. This file stores the $V_{rs}^{k\tau}(abcd)$ coefficients.

$i_2$**:** Unit number of the MCBP INPUT FILE. This is an MCBP OUTPUT FILE from an incomplete run.

When $i_1 = i_2$ the file is the MCBP INPUT/OUTPUT FILE.

---

### 3.2.31 BENA Invocation and Option Specification Card

Format: `BENA` $\{0$–$25\}i$

The default is to print eigenvalues and only the $jj$-basis eigenvectors. The following options over-ride the default.

**1** : Diagonalise $\boldsymbol{H}^{\mathrm{Transv}}$ on its own, then $\boldsymbol{H}^{\mathrm{Transv}} + \boldsymbol{H}^{\mathrm{QED}}$. [Diagonalise the transverse interaction matrix to obtain the new Coulomb+transverse ASF basis. Rediagonalise the hamiltonian after computing the QED corrections.]

**2** : Diagonalise $\boldsymbol{H}^{\mathrm{Transv}}$ on its own, then estimate QED contributions as the diagonal elements of $\boldsymbol{H}^{\mathrm{QED}}$. [Diagonalise the transverse interaction matrix to obtain the new Coulomb+transverse ASF basis. Do not rediagonalise the hamiltonian after computing the QED corrections; note that only the diagonal contributions are included if this is the case.]

**3** : Diagonalise $\boldsymbol{H}^{\mathrm{Transv}}$ on its own; do not estimate QED contributions.

**5** : Print out contributions of $H_{rs}^{\mathrm{Transv}}$ to eigenvalues.

**7** : Make use of user-supplied routine VACUSR for evaluating $V^{\mathrm{VP}}(r)$.

**8** : Use point nucleus $V^{\mathrm{VP}}(r)$.

**9** : Use second-order $V^{\mathrm{VP}}(r)$ only.

**11** : Give information on evaluation of $\phi_k(\omega r)$ and $\psi_k(\omega r)$ in routine BESSEL.

**12** : Print $\boldsymbol{H}^{\mathrm{Transv}}$ before transformation to eigenvector basis of $\boldsymbol{H}^{\mathrm{DC}}$.

**13** : Print restart information.

**14** : Print out each contribution to $H_{rs}^{\mathrm{Transv}}$.

**15** : Print vacuum polarisation integrals and self-energy for each orbital; print QED contributions to CSF manifold.

**16** : Print matrix before adding diagonal terms and subtracting average energy from diagonal. [Print out H(QED) in the Breit basis.]

**18** : Print matrix before diagonalising.

**19** : Print $V^{\rm VP}(r)$.

**21** : Print eigenvectors in new basis. [Transform eigenvectors to $LS$-basis and print.]

**22** : Do not print eigenvectors in standard basis. [Do not print eigenvectors in $jj$-basis.]

**23** : Do not print eigenvectors in any basis. [Do not print any eigenvectors.]

**24** : Print eigenvectors in $\boldsymbol{c}^{\rm DC}$ basis.

**25** : Do not print eigenvalues.

**26** : Print neither eigenvalues nor eigenvectors.

**27** : Print summary in Ryd.

**28** : Print summary in $\rm cm^{-1}$.

**29** : Print summary in eV.

**30** : Do not print final summary. [Do not call routine SUMMRY to obtain final summary of contributions to energy levels.]

---

### 3.2.32 BENA Unit Specification Card

Format: $\{3\text{–}5\}i$

$i_1$**:** Unit number of the SORTED MCP (OUTPUT) FILE. See the MCP Invocation and Unit Specification CARD.

$i_2$**:** Unit number of the MCDF (DUMP or READ/DUMP) FILE from a *converged* run. See the MCDF Unit Specification Card.

$i_3$**:** Unit number of the MCBP (OUTPUT or INPUT/OUTPUT) FILE. See the MCBP Invocation and Unit Specification Card.

$i_4$**:** Unit number of the BENA OUTPUT FILE. This file stores the $S^{k\tau}(abcd)$ integrals.

$i_5$**:** Unit number of the BENA INPUT FILE. This is a BENA OUTPUT FILE from an incomplete run.

If $i_4 = i_5$ this is the BENA INPUT/OUTPUT FILE.

---

### 3.2.33 PRINT Card

Format: `PRINT` $\{0$–$n_c\}i$

$i_k$**:** Print information related to this level. The levels are ordered by increasing energy.

If no levels are specified, information is printed for all $n_c$ levels.

The default is printing for all $n_c$ levels for `AL` or `EAL` calculations, $i$ for `OL` and $i(j)$ with $j = 1, \ldots, n_L$ for `EOL` calculations.

---

### 3.2.34 LOW1 Card

Format: `LOW1` $n$

$n$**:** Factor reducing the transverse photon frequency to calculate the $\omega \rightarrow 0$ limit of the transverse Breit interaction. $n =$ `1E-3` is recommended. The default value is `1`.

---

### 3.2.35 MCT Invocation and Unit Specification Card

Format 1: `MCT` $i_1$ $i_2$ $i_3$ $i$'s

or

Format 2: `MCT` $i_1$ $i_2$ $i_3$ `0` $i$'s

$i_1$**:** Unit number of the UNSORTED MCT OUTPUT FILE. This file stores the CSF-manifold-based list of $d_{rs}^{\pi k}(ab)$ coefficients.

$i_2$**:** Unit number of the SORTED MCT OUTPUT FILE. This file stores the integral-based list of $d_{rs}^{\pi k}(ab)$ coefficients.

$i_3$**:** Unit number of the UNSORTED MCT INPUT FILE. This is an UNSORTED MCT OUTPUT FILE from an incomplete run. Set to `0` if not present.

If $i_1 = i_3$ this is the UNSORTED MCT INPUT/OUTPUT FILE.

$i$**s:** Integers specifying the ranks, $k$, and parities, $\pi$, of the operators. A positive (negative) integer implies even (odd) parity if Format 1 is used. If Format 2 is used, the $i \geq 0$ specify the ranks, $k$, of the operators and *both* parities are implied.

---

### 3.2.36 OSCL Invocation and Option Specification Card

Format: `OSCL` $\{0\text{–}17\}i$

$i$:

**1** : Use $E_\Gamma^{\rm DC}$, $\boldsymbol{c}_\Gamma^{\rm DC}$.

**2** : Use $E_\Gamma^{\rm DC+Transv+QED}$, $\boldsymbol{c}_\Gamma^{\rm DC+Transv+QED}$.

**3** : Both 1 and 2.

**4** : Use $E_\Gamma^{\rm DC+Transv}$, $\boldsymbol{c}_\Gamma^{\rm DC+Transv}$, then use $E_\Gamma^{\rm DC+Transv+QED}$, $\boldsymbol{c}_\Gamma^{\rm DC+Transv+QED}$.

**5** : Print transition probabilities in au.

**6** : Sort transitions in ascending order of energy.

**7** : Print transition wavelengths in Å.

**8** : Print transition energies in $\rm cm^{-1}$.

**9** : Print transition energies in eV.

**10** : Print transition frequencies in Hz.

**12** : Print $\bar{M}_{ab}^{\rm e,l,m}$.

**13** : Print the integrand of $\bar{M}_{ab}^{\rm l}$.

**14** : Print $I_L^\pm$, $J_L$.

**15** : Print the integrands of $I_L^\pm$, $J_L$.

**16** : Print $j_L(\omega r/c)$ as evaluated in routine BESSJ.

**18** : Print out $d_{rs}^{\pi k}(ab)$ after they are read in.

**19** : Write out $E_\Gamma$, $\boldsymbol{c}_\Gamma$.

---

### 3.2.37 OSCL Unit Specification Card

Format: $\{2\text{–}3\}i$

$i_1$**:** Unit number of the MCDF (DUMP or READ/DUMP) FILE from a converged run. See the MCDF Unit Specification Card.

$i_2$**:** Unit number of the SORTED MCT (OUTPUT) FILE. See the MCT Invocation and Unit Specification Card.

$i_3$**:** Unit number of the OSCL OUTPUT FILE. This file stores transition data.

---

### 3.2.38 LEV Card

Format: `LEV` $\{2\text{–}n_\mathrm{c}\}i$

$i_k$**:** Sequence number (in ascending order of energy) of levels between which transition data is to be *printed* (it is calculated for all levels).

The default is all $n_\mathrm{c}$ levels for `AL` or `EAL` calculations, none for `OL` calculations, and all $j = 1, \ldots, n_\mathrm{L}$ levels $i(j)$ for `EOL` calculations.

---

### 3.2.39 CUT Card

Format: `CUT` $n$

$n$**:** Smallest transition rate (in $\mathrm{s}^{-1}$) to be *printed*. The default value is `0`.

---

### 3.2.40 LOW2 Card

Format: `LOW2` $n$

$n$**:** Factor reducing the photon frequency to calculate the $\omega \to 0$ limits of the transition matrix elements. $n =$ `1E-3` is recommended. The default value is `1`.

---

### 3.2.41 END Card

Format: `END`

---

## 3.3 Dimensions

The dimensions are set by the file `grasp2.inc`.

Numerical Plants: These are assigned numerical values with equivalences of the type `KEY = 73`. All numerical plants must be assigned integer values $\geq 1$ except as specified below.

| | |
|---|---|
| `C3` | The maximum number of items in a single input card (including continuation lines but not `C`s) to CARDIN: max $(2 \cdot \mathtt{NW} + 1,\ 2 \cdot \mathtt{NC} + 2)$. |
| `KEY` | `INT(2**((IL-1)/5))`, where `IL` is the length of the integer in bits. (Usually `73`.) |
| `LL` | The maximum number of integral labels allowed. |
| `LM` | The maximum number of Lagrange multipliers: $\mathtt{LM} \leq (\mathtt{NW} \cdot (\mathtt{NW} - 1))/2$. |
| `LV` | The maximum number of pairs of levels between which transitions are to be calculated: $\mathtt{LV} = (\mathtt{NC} \cdot (\mathtt{NC} - 1))/2$. |
| `MC` | The number of nonrelativistic configurations. |
| `MW` | The number of nonrelativistic $(nl)$ orbitals. |
| `NB` | The total number of $S^{k\tau}(abcd)$ integrals. |
| `NC` | The number of relativistic CSF manifolds. |
| `ND` | The maximum value of $N_a^z$. |
| `NJ` | The maximum number of orbital $j$ values allowed. |
| `NLP` | The maximum number of lines on a page of printout; default is `70`. |
| `NM` | The total number of $V_{rs}^k(abcd)$ and $T_{rs}(ab)$ coefficients. |
| `NP` | The number of points in the radial grid: $\mathtt{NP} \geq 220$. |
| `N1` | $\mathtt{N1} = \mathtt{NP} + 10$. |
| `NRP` | The maximum number of rank/parity combinations for one-electron operators. |
| `NS` | The maximum number of $d_{rs}^{\pi k}(ab)$ coefficients of a given rank and parity. |
| `NT` | The maximum number of $d_{rs}^{\pi k}(ab)$ coefficients. |
| `NV` | The number of valence orbitals (those subject to a core-polarisation potential). |
| `NW` | The number of relativistic $(nlj)$ orbitals. $\mathtt{NW} \leq \mathtt{KEY}$. |
| `NX` | The maximum value of $N_a^x$. |
| `NY` | The maximum value of $N_a^y$. |
| `OL` | The maximum value of $n_{\mathrm{L}}$: $\mathtt{OL} < \mathtt{NC}$. |
| `OP` | The maximum number of open nonrelativistic subshells: this must be `4` unless the user has modified GRASP$^2$ according to the comment cards in routine DATNR. |
| `2OP` | $\mathtt{2OP} = 2 \cdot \mathtt{OP}$. |
| `PR` | $\mathtt{PR} = (\mathtt{NC} \cdot (\mathtt{NC} + 1))/2$ if option 27 on the MCDF Invocation and Option Specification Card is to be used, `1` otherwise. |

# Chapter 4

# GRASP$^2$ routines

The routines in GRASP$^2$ are described in alphabetical order. Routines that are part of the NJGRAF package are separate.

**Routine GRASP**

This is the entry point into the package. After performing some setup, it directs control to each section in turn. After processing one problem it loops back indefinitely to start a new problem.

Input: IRD=5, FILE='GRASP2.INP', STATUS='OLD'
Output: IPD=6, FILE='GRASP2.OUT', STATUS='UNKNOWN'

Routine(s) called: BENA, CARDIN, DATAIN, (ERRSET), FACTT, MCBP, MCDF, MCP, MCT, TIMER, OSCL.

**TIME card — COMPULSORY**

Format: string of 1 to 4 characters, 1 integer

Label "TIME" (minimal abbreviation "T") (compulsory): the maximum CPU time for this run follows on this card.

1 integer (COMPULSORY): CPU time in seconds for this problem

**Routine ARCTAN—function**
`(ARG1,ARG2)`

ARCTAN = $\tan^{-1}$(`ARG1`/`ARG2`) with $0 \leq$ ARCTAN $< 2\pi$

**Routine AVCONF**
`(IPD,JCFN,NST,NCSF,IPAR)`

**Routine BENA**
`(IPD,IMCP,IDUMP,IMCBP,ISTOUT,ISTIN)`

This routine controls the main sequence of routine calls for the calculation of the transverse Breit and QED corrections to the MCDF energy levels.

Routine(s) called: BREMAT, CHEKFL, CLOSFL, ENGOUT, LOAD, MATOUT, NEWBAS, OPENFL, ORDER, QED, SUMMRY.

**Routine BESSEL**
`(IPD,IA,IB,IK,IW,K)`

This routine evaluates the functions

$$\begin{aligned} \text{BESSJ} &= \frac{(2k+1)!!}{(\omega_{ab}\, r)^k}\, j_k(\omega_{ab}\, r) - 1 \\ &= \phi_k(\omega_{ab}\, r) - 1 \end{aligned} \tag{4.1}$$

and

$$\begin{aligned} \text{BESSN} &= -\frac{(\omega_{ab}\, r)^{k+1}}{(2k-1)!!}\, n_k(\omega_{ab}\, r) - 1 \\ &= \psi_k(\omega_{ab}\, r) - 1 \end{aligned} \tag{4.2}$$

where $j_k$ and $n_k$ are spherical Bessel functions, and

$$\omega_{ab} = \frac{|E_a - E_b|}{c} \tag{4.3}$$

where $E_i$ is the eigenvalue for orbital $i$.

The routine uses equations from Abramowitz and Stegun (1965 [1]) to evaluate the functions. Devices are used to reduce the number of actual evaluations of these functions. The writeup, McKenzie et al (1980 [21]), is incorrect in its description of the output of this routine.

Routine(s) called: none

**Routine BESSJ**
`(IPD,W)`

This routine evaluates Bessel functions $j_k\left(\frac{\omega r}{c}\right)$ at the grid points for `K=L-1,L,L+1` and stores them in the arrays `BJ(..,1)`, `BJ(..,2)`, `BJ(..,3)` respectively. It uses a power series expansion for small $r$ and switches to SIN/COS expansion when more than 4 terms in the power series are required.

Routine(s) called: none

**Routine BREID**
`(IPD,IBUG1,IBUG2,JA1,IPCA,JB1)`

This routine computes closed shell contributions - `AAAA` and exchange only.

Routine(s) called: CLRX, CXK, ITRIG, TALK, SNRC.

**Routine BREIT**
`(IPD,IBUG1,IBUG2,IBUG3,JA1,JB1,JA2,JB2)`

This routine computes the coefficients appearing in Grant and McKenzie (1980 [16]). The coefficients for each choice of orbitals `JA1`, `JB1`, `JA2`, `JB2` depend on 2 further parameters `NU` and `K`; there are `IMU` integrals for each such choice, where:

| ITYPE | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| IMU | 4 | 8 | 1 | 1 | 3 | 4 |

Routine(s) called: CXK, GENSUM, ITRIG, KNJ, LTAB, MODJ23, MUMDAD, NJGRAF, OCON, TALK, SETJ, SNRC

An outline of the calculation follows.

**Step 1.0** Initialise pointers and flags and set any tables required.

In this segment, the array `IS` points to the full list of orbitals, the array `JS` to the array `JLIST` of peel orbital pointers.

1.1 Initialisation

**Step 2.0** Set quantum numbers of spectator shells.

**Step 2.1** Examine spectator shells for orthogonality

**Step 3.0** Start main calculation, begin with common factors

**Step 3.1** Set range of tensor index `NU`

**Step 3.2** Set parameters of summation over parent (barred) terms. The array `IROWS` is formed to point to the list of allowed parents of active shells in the array `NTAB`.

**Step 4.0** Sum over all parent terms permitted by angular momentum and seniority selection rules

Treat `IA1` = `IB1` as a special case

Treat `IA2` = `IB2` as a special case

At this point, the current parent has been completely defined, and its quantum numbers can now be set. The `JTQ` arrays must be set if `IA1` = `IB1` or `IA2` = `IB2`. The matrix element should be diagonal in barred quantum numbers.

**Step 4.1** Evaluate product of 4 CFPs

**Step 4.2** Set arrays for defining the recoupling coefficient

Set up the arrays and variables for the direct case.

Set up the arrays and variables for the exchange case.

**Step 4.3.1** Summation for direct terms

**Step 4.3.2** Summation for exchange terms

**Step 4.4** Insert outside factors

**Step 5.0** Output results

**Step 6.0** Fault diagnostic prints

**Routine BREMAT**
`(IPD,IMCBP,IDUMP,ISTOUT,ISTIN,NREC)`

This routine computes the transverse interaction matrix in the Coulomb ASF basis. This is done in two steps: first the matrix is generated in the CSF basis; next it is transformed to the Coulomb ASF basis using the matrix of eigenvectors of the Coulomb hamiltonian matrix.

In generating the transverse interaction matrix in the CSF basis, for each pair of configurations (`JA`,`JB`) it reads the relevant MCBP coefficients, and checks the integral labels against a sorted list to determine whether the integral has been calculated. If it has, the integral is taken from the list. If not, it is evaluated. The integral is then multiplied by the relevant coefficient and added to the matrix element. When each row of the matrix has been calculated, it is written on to the end of the MCDF dump file, so that if the calculation runs out of time, it can be resumed without much loss of effort.

Routine(s) called: BESSEL, BRRA, MATOUT.

**Routine BRRA**
`(IPD,ITYPE,IA,IC,IB,ID,K,RESULT)`

This routine evaluates the Breit interaction integrals:

- `ITYPE` = 1: general $R(k; ac|bd)$
- `ITYPE` = 2: general $S(k; ac|bd)$
- `ITYPE` = 3: $R(k; a, bd)$
- `ITYPE` = 4: $F(k; a, b)$
- `ITYPE` = 5: $G(k; a, b)$
- `ITYPE` = 6: $H(k; a, b)$

Routine(s) called: RKINT, SKINT.

**Routine CARDIN**
`(IRD,IPD,NN,RT,IT,JT,LT,CT,KT,ND)`

This routine reads a logical record (which may extend over more than one card or physical record) in free format, and does some processing of the data on the record. Checks are made on the consistency of the data. See the errors section for a description of errors which will terminate the program. Numbers are read by constructing a FORMAT statement and reading from a string.

Description of parameter list:

- NN : number of data items which have been read
- RT : array containing real number values of data where appropriate; all integers are also stored as real numbers; RT is zero for other than numeric data values
- IT : array containing integer values; real numbers are integerised by truncation and stored in IT if they are not too large, except when division (/) is used, in which case it contains the integerised numerator. When an orbital is being specified, it contains the principal quantum number $n$; it is zero if a string or a pure logical value is given
- JT : array containing integer values; JT is zero unless division is used, when it contains the integerised denominator, or when an orbital is being specified, when it contains the $\kappa$ quantum number
- LT : array containing logical values; LT is .FALSE. unless a logical value of .TRUE. (or T or TRUE) is given, or a left parenthesis ( is given which is not part of a repeat count

- CT : array containing strings; CT is blank unless a string is given or an orbital is specified; in the latter case CT contains the orbital angular label, e.g. P- for $l = 1$, $j = 1/2$; valid strings must commence with a letter, but may contain numbers and the characters / . and - as well as letters
- KT : number of non-blank characters in CT
- ND : sets dimension of above arrays

The following characters are recognised by CARDIN:

- all letters and numbers and + - * / ( ) and .
- ' ' (blank) is used to separate data items
- 'C' can be used to continue a logical record onto the next physical record; when this character is found in isolation, the remainder of the current record is checked for data — an error occurs if any is found — and a new record is read; there is no limit on the number of continuations
- '/' can be used to input fractions with numerator or denominator being real or integer
- 'E' denotes exponent (of 10); 'D' may also be used; the exponent must be an integer between +ITENMX and -ITENMX (see main program for definition)
- '*' is used to indicate a repeated value or series of values (as in FORTRAN DATA statements). If a series of values is used, they must be enclosed in parentheses e.g. 4*(1 2 3) is used for 1 2 3 1 2 3 1 2 3 1 2 3; values may be numeric or orbitals, and may include a range (see '-' below)
- '-' in isolation is used to specify a range of values, e.g., 4 - 7 is used for 4 5 6 7 ; may be used for numbers and orbitals; in the latter case, the logical value for the first orbital is set to .TRUE. and no further processing of the range is done, whereas in the former case, the series of numbers is developed in situ
- 'T' in isolation denotes logical value .TRUE.
- 'F' in isolation denotes logical value .FALSE.
- '(' is used in conjunction with * for repeating a group of numbers (see '*' above); otherwise it sets a logical value of .TRUE. for the following number
- ')' terminates a group of numbers (see '*' above); otherwise it sets a logical value of .FALSE. for the preceding number

Routine(s) called: LNGTH.

**Routine CFOUT**
`(IPD)`

This routine prints configuration state function data in the usual format.

Routine(s) called: none

**Routine CFP**
`(IPD,LOCK,NEL,IJD,IVD,IWD,IJP,IVP,IWP,COEFP)`

This routine selects the appropriate table of fractional parentage coefficients in $jj$-coupling.

Input variables:

- `IPD` : output stream number
- `LOCK` : $\pm(2j+1)$
- `NEL` : number of equivalent electrons in shell
- `IJD/IJP` : total $J$ of daughter/parent state
- `IVD/IVP` : seniority of daughter/parent state
- `IWD/IWP` : other quantum number (if needed)

Output variable:

- `COEFP` : numerical result

This control routine does not check the input variables for consistency, except the trivial case of $j = 1/2$. All other checks are performed at a lower level. The package will return correct results for $j = 3/2, 5/2, 7/2$. Higher values of $j$ return a value 1.0 if `NEL` $= 1$ or 2; otherwise 0 with an error signal.

Routine(s) called: CFP3, CFP5, CFP7, CFPD.

These routines have been taken from Grant (1972 [10]).

The following changes have been made:

- correction deck has been incorporated
- routine CFP has been modified
- programs now stops when an error is detected

**Routine CFP3**
`(IPD,NEL,IJD,IJP,COEFP)`

Table look-up for fractional parentage coefficients of equivalent electrons with $j = 3/2$. See listing of CFP for argument list.

Routine(s) called: none

**Routine CFP5**
`(IPD,NEL,IJD,IVD,IJP,IVP,COEFP)`

Table look-up for fractional parentage coefficients of equivalent electrons with $j = 5/2$. See listing of CFP for argument list.

Routine(s) called: none

**Routine CFP7**
`(IPD,NEL,IJD,IVD,IJP,IVP,COEFP)`

Table look-up for fractional parentage coefficients of equivalent electrons with $j = 7/2$. See listing of CFP for argument list.

Routine(s) called: none

**Routine CFPD**
`(IPD,LOCK,NEL,IJD,IVD,IWD,IJP,IVP,IWP,COEFP)`

This is a dummy routine. It returns correct values for 1 or 2 particle or single hole states, and signals an error otherwise.

Routine(s) called: none

**Routine CGAMMA**
`(IPD,ARGR,ARGI,RESR,RESI)`

This routine returns in `RES` the complex *Gamma* function of the complex argument `ARG`. The real and imaginary parts of `RES` and `ARG` are distinguished by the suffixes R and I respectively.

Only `RESR` is nonzero if `ARGI` is zero.

The ARCTAN function required must return angles (in radians) in the range $[0, 2\pi)$.

Routine(s) called: ARCTAN.

**Routine CHEKFL**
`(IPD,NUNIT,GTYPE,GSTAT)`

This routine checks the standard GRASP file header.

`NUNIT` is the file unit number.

`GTYPE` is the GRASP file type.

`GSTAT` is the GRASP file status.

The check on the file status can be overridden by specifying it as a blank. The check on the basic dimensions is not made on orbitals files.

**Routine CLOSFL**
`(NUNIT)`

This routine closes files. The STATUS is always assumed to be 'KEEP'.

**Routine CLRX—function**
`(KAPPAA,K,KAPPAB)`

The value of CLRX is the $3j$-symbol:

$$\begin{pmatrix} j_a & k & j_b \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{pmatrix} \tag{4.4}$$

The KAPPAA and KAPPAB are $\kappa$ angular quantum numbers. The formula is taken from Brink and Satchler (1993 [4]).

The logarithms of the first MFACT factorials must be available in common block /FACTS/ for this program to function correctly. Note that $N!$ is stored in FACT(N+1).

Routine(s) called: none

**Routine CONSIS**
(J)

This routine computes the weighted self-consistency of orbital J.

Routine(s) called: none

**Routine CONVRT**
(NUMBER,CHRSTR,NDIGIT)

This routine accepts an integer NUMBER as input. The output CHRSTR is a character string in whose first NDIGIT places the integer is written as the corresponding character string.

Routine(s) called: none

**Routine COR**
(IPD,IBUG1,IBUG2,IBUG3,JA1,JB1,JA2,JB2)

This routine computes the MCP coefficients. Equation numbers are those of Grant (1973 [11]).

Routine(s) called: CRE, GENSUM, ITRIG, LTAB, MODJ23, MUMDAD, NJGRAF, OCON, SETJ, SKRC, SPEAK, KNJ

An outline of the calculation follows.

- 1.0 Initialise pointers and flags and set any tables required.

  In this segment, the array IS points to the full list of orbitals, the array JS to the array JLIST of peel orbital pointers.

- 1.1 Initialisation

- 2.0 Set quantum numbers of spectator shells.

- 2.1 Examine spectator shells for orthogonality

- 3.0 Start main calculation, begin with common factors

- 3.1 Set range of tensor index K

- 3.2 Set parameters of summation over parent (barred) terms in eq.(5) (loc cit). The array IROWS is formed to point to the list of allowed parents of active shells in the array NTAB.

- 4.0 Sum over all parent terms permitted by angular momentum and seniority selection rules

  Treat IA1=IB1 as a special case

Treat IA2=IB2 as a special case

At this point, the current parent has been completely defined, and its quantum numbers can now be set. The JTQ arrays must be set if IA1=IB1 or IA2=IB2. The matrix element should be diagonal in barred quantum numbers.

- 4.1 Evaluate product of 4 CFPs
- 4.2 Set arrays for defining the recoupling coefficient

  Set up the arrays and variables for the direct case.

  Set up the arrays and variables for the exchange case.
- 4.3 Calculate AD, eq.(6), without the phase factor.
- 4.4 Calculate AE, eq.(6), without the phase factor.
- 4.5 Insert factors independent of barred quantum numbers.

  Begin with common statistical factors, eq.(5).
- 4.6 Compute products of reduced matrix elements, eq.(7).
- 5.0 Output results
- 6.0 Fault diagnostic prints

**Routine CORD**
(IPD,IBUG1,IBUG2,JA1,IPCA,JB1)

This routine computes the MCP coefficients for contributions involving closed shells. The standard formulae are given in Grant (1970 [9]) eq.(8.33).

In this segment JA1, JB1 point to the JLIST array, IA1, IB1 to the full list of orbitals.

Routine(s) called: CLRX, SPEAK.

**Routine COUNTX**
(FR,MTPFR,NNCFF,SGN)

This routine counts the nodes in the radial function FR using the criteria given by Froese Fischer (1986 [7]) p.314-315. The function RF is assumed defined on the first MTPFR points of the radial grid. The sign of the function at the first oscillation is also determined.

Routine(s) called: none

**Routine CPPOT**
(IPD)

This routine tabulates effective nuclear potential (the sum of the nuclear and core polarisation potentials) for all valence orbitals. The Norcross-Seaton form of the CUTOFF is used.

Routine(s) called: WFN

**Routine CRE—function**
`(KAP1,K,KAP2)`

This routine computes the relativistic reduced matrix element

$$(j_1||C(K)||j_2) = (-1)^{j_1+\frac{1}{2}} \sqrt{(2j_1+1)(2j_2+1)} \begin{pmatrix} j_1 & K & j_2 \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{pmatrix} \tag{4.5}$$

defined by eq.(5.15) in Grant (1970 [9]) p.762.

`KAP1`, `KAP2` are the $\kappa$ values corresponding to $j_1$, $j_2$. The triangle conditions are tested by routine CLRX.

Routine(s) called: CLRX.

**Routine CSFM**
`(IPD,ASFA,ASFB,LEV1,LEV2)`

This routine calculates the CSF Coulomb, Babushkin and magnetic matrix elements for a transition between levels separated by energy `OMEGA`.

Routine(s) called: SPME.

Option 18 (LTC-CSFM) — Print MCT coefficient.

**Routine CSFOUT**
`(IPD,LHEAD)`

This routine prints the CSF in the given coupling scheme. For each CSF, An output line is constructed which consists of three strings, filled with the symbols used to specify the appropriate occupation numbers, angular momenta, etc, for each open subshell and the way they are coupled.

Description of the parameter list:

`LHEAD`: option to select printing of heading for each nonrelativistic configuration. Set to .TRUE. if heading is required.

Routine(s) called: LNGTH.

**Routine CXK**
`(IPD,S,IS,KAPS,NU,K,IBR,IEX)`

This routine computes the coefficients of radial integrals in the expansion of the effective interaction strength:

$$X(k; a, b, c, d). \tag{4.6}$$

Input variables:

- `IS` : orbital labels
- `KAPS`: values of $2\kappa$

- NU : order of radial integral
- K : index of tensor operator
- IBR : classifies type of radial integral
- IEX : 1 for direct, 2 for exchange terms

There are 4 distinct cases:

- IBR = 1
  A. all states distinct
  B. $((a = b)$ .AND. $(c \neq d))$ or $((a \neq b)$ .AND. $(c = d))$
  these give 12 distinct radial integrals, with values of K and NU limited only by angular momentum and parity
- IBR = 2
  $((a = c)$ .AND. $(b \neq d))$ or $((a \neq c)$ .AND. $(b = d))$
  this case gives 1 non-zero integral when K = NU is odd
- IBR = 3
  $((a = c)$ .AND. $(b = d))$ and $(a \neq b)$
  integrals of magnetic $F$-type when K = NU is odd
- IBR = 4
  $((a = d)$ .AND. $(b = c))$
  gives 3 magnetic $G$-type integrals and 4 $H$-type integrals

Output:

S : coefficients S(MU) MU = 1,12

Routine(s) called: CRE.

**Routine DACON**

This routine includes the contribution from the off-diagonal $I(a, b)$ integrals in the 'exchange' term.

Routine(s) called: DPBDT

**Routine DAMPCK**
(IPR,J,ED1,ED2)

This routine determines the damping factor appropriate to the present orbital. The algorithm is taken from Froese Fischer's program MCHF.

**Routine DAMPOR**
(IPD,J,INV)

This routine damps the orbital wavefunction with index J. It also stores the previous determination of this orbital.

Routine(s) called: COUNTX, RINT.

**Routine DATAIN**
(IRD,IPD,IBUG,IO,IOMCDF)

Only the TIME card is read by PROGRAM GRASP. The present routine controls the reading of the remainder of the command/data file.

Routine(s) called: CALEN, CARDIN, DATSCF, DATR, DATNR, LNGTH.

- **Title card — COMPULSORY**

  Format: 1 string (1 to 72 alphanumeric characters)

  This is the only card which is not read by CARDIN.

- **Convention/dimension/transformation/print-control card — COMPULSORY**

  Format 1: 2 or 3 integers

  Format 2: 3 integers and up to 5 labels

- **MCP invocation and input/output stream specification card — OPTIONAL**

  Format: string of 3 characters, 2 to 3 integers

- **MCDF invocation and options specification card — OPTIONAL**

  Format: string of 4 characters, 0 – 34 different integers

- **MCT invocation and input/output stream specification card — OPTIONAL**

  Format 1: string of 3 characters, 3 integers, 1 – NRP integers
  Format 2: string of 3 characters, 3 integers, 0, 1 to NRP/2 positive integers

- **MCBP invocation and input/output stream specification card — OPTIONAL**

  Format: string of 4 characters, 1 or 2 integers

- **BENA invocation and options specification card — OPTIONAL**

  Format: string of 4 characters, 0 – 25 integers

- **BENA input/output stream specification card — CONTINGENT**

  Format: 3 – 5 integers

- **LOW1 card — OPTIONAL**

  Format: string of 4 characters, 1 real number

- **PRINT card — OPTIONAL**

  Format: string of 1 to 5 characters, 0 to NCF integers

- **OSCL invocation and options specification card — OPTIONAL**

  Format: string of 4 characters, 0 – 17 integers

- **OSCL input/output stream specification card — CONTINGENT**

  Format: 2 – 3 integers

- **LEV card — OPTIONAL**

  Format: string of 3 characters, 2 to [NCF] integers

- **CUT card — OPTIONAL**

  Format: string of 3 characters, 1 real number

- **LOW2 card — OPTIONAL**

  Format: string of 4 characters, 1 real number

- **END card — COMPULSORY**

  Format: string of 3 characters

**Routine DATANG**
`(IRD,IPD,RT,IT,JT,LT,CT,KT,ND)`

Read angular data using the free format card reader CARDIN.

**Coupling scheme cards — CONTINGENT**

Format: 1 integer, integers or half-integers, or pairs of integers or half integers and integers, integers or half- integers

**Routine DATNR**
`(IRD,IPD,RT,IT,JT,LT,CT,KT,ND,IOP,IBUG)`

This routine reads card input defining nonrelativistic configurations and uses it to generate all possible relativistic configurations which produce the required total $J$ value. `NCFN` is the number of nonrelativistic configurations to be defined; `NWM` is the number of nonrelativistic orbitals to be defined; and `ND` is the maximum number of numbers that can be read in CARDIN.

Routine(s) called: CARDIN, JJCSF, REORDR, JJRECP, JRECUP, JJJLSJ, LLLSSS, JLKSJ, AVCONF, MANOUT.

**Orbital data cards; [NW] in number — COMPULSORY**

Format: label, 0 or 1 or [NW] integers

label: this consists of an integer [NP(I)] and a spectroscopic angular-momentum label [NH(I)] for the orbital [I]; no intervening blanks must be present; the labels are the usual ones: "S", "P", "D", etc.

0 or 1 or [NW] integers — optional — occupation numbers for orbital [I]; if 0 numbers are input, the subshell is assumed full in every configuration; if 1 integer is given, each

configuration has this occupation number; if [NCF] numbers are present, these are the subshell occupation numbers for each configuration

**ANG card — COMPULSORY**

See the description of this card in routine DATANG

**Angular-momentum coupling information card — COMPULSORY**

[NCF] in number

Format: 1 to NOPEN+4 integers and/or logical values

1 integer — COMPULSORY — total $J$ value for the nonrelativistic configuration; if $-1$, all possible J values are generated

1 integer — OPTIONAL — number of subshells to be transformed to $LSJ$ coupling

1 integer — OPTIONAL — number of subshells to be transformed from $LSJ$ to $LLSS$ coupling

1 logical value — OPTIONAL — flag to indicate whether $J - L$ coupling is to be used on the last $LSJ$ subshell ("F" or "FALSE" = no, "T" , "TRUE" or "J-L" = yes)

Remaining values — OPTIONAL — a list of the subshells in order of coupling. Parentheses may be used to indicate that a pair of subshells is to be coupled together before coupling to the resultant.

**Routine DATOUT**
`(IPD)`

This routine outputs some information read from the MCDF dump.

Routine(s) called: none

**Routine DATR**
`(IRD,IPD,RT,IT,JT,LT,CT,KT,NMAN,NWM,ND,IBUG)`

This routine reads the card input defining relativistic configurations as described in the CPC write up.

`NMAN` : number of relativistic configurations to be defined
`NWM` : number of relativistic orbitals to be defined
`ND` : maximum number of numbers that can be read in CARDIN

Routine(s) called: CARDIN, CONVRT, DATANG, LNGTH.

**Orbital data cards; [NW] in number — COMPULSORY**

Format: string, 0, 1 or [NCF] integers

**ANG card — COMPULSORY**

Format: string of 3 characters, 0 to 7 integers

**Routine DATSCF**
`(IRD,IPD,RT,IT,JT,LT,CT,KT,ND,IOP,IOMCDF)`

This routine gathers the basic input data for the MCDF and SCF packages.

Routine(s) called: CONVRT, LNGTH

- **MCDF file specification card — COMPULSORY**

  Format: 3 to 5 integers

  `IOMCDF(1)` — `IR1`> 0 — required — 'Sorted MCP file'

  `IOMCDF(2)` — `IR2`> 0 — required — 'MCDF read file'

  `IOMCDF(3)` — `IP2`> 0 — required — 'MCDF dump file' — can be set equal to `IR2`

  `IOMCDF(4)` — `IR3`> 0 — required — 'ORBOUT read file'

  `IOMCDF(5)` — `IP3`> 0 — required — 'ORBOUT dump file' — can be set equal to `IR3`

- **Nuclear charge/mass card — COMPULSORY**

  Format: 1 to 2 real numbers

  `Z` — nuclear charge — required

  `ATW` — nuclear mass (defaults to zero)

- **NUCLEUS card — OPTIONAL**

  Format: string of 7 characters, string of 5 characters, 1 to 2 real numbers

- **FIX card — OPTIONAL**

  Format: string of 3 characters, 1 to [NW]-1 integers

  This card must precede the ORBOUT card if both are used

- **LOAD card — OPTIONAL**

  Format: string of 4 characters, [NW] integers

- **ORBOUT card — OPTIONAL**

  Format: string of 5 characters, 0 to [NW] integers

- **RHFX card — OPTIONAL**

  Format: string of 4 characters, 1 to 3 real numbers

- **SCF card — OPTIONAL**

  Format: string of 3 characters, 1 to 5 pairs of keywords and numbers

- **GRID card — OPTIONAL**

  Format: string of 4 characters, 1 to 4 pairs of keywords and numbers

- **RLDA card — OPTIONAL**

  Format: string of 4 characters, 0 or 1 integer, 1 real number

- **PRINT card — OPTIONAL**

  Format: string of 5 characters, 0 to NCF integers

  This card may be input only once, either in this section or in DATAIN

- **SCREEN card — OPTIONAL**

  Format 1: string of 6 characters, 1 real number

  Format 2: string of 6 characters, 1 to [NW] pairs of 1 integer and 1 real number

- **METHOD card — OPTIONAL**

  Format 1: string of 2 to 6 characters, 1 integer

  Format 2: string of 2 to 6 characters, 1 to [NW] pairs of integers

- **CDAMP card — OPTIONAL**

  Format 1: string of 2 to 5 characters, 1 real number

  Format 2: string of 2 to 5 characters, 1 to [NCMIN] pairs of 1 integer and 1 real number

- **ODAMP card — OPTIONAL**

  Format 1: string of 2 to 5 characters, 1 real number

  Format 2: string of 2 to 5 characters, 1 to [NW] pairs of 1 integer and 1 real number

- **CPOLPOTL card — OPTIONAL**

  Format: string of 8 characters, 1 integer

  N.B. This card must be specified after the FIX card

- **Core-polarisation potential data cards — CONTINGENT**

  Read orbital, dipole polarisability, cutoff radius

- **NOINVERT card — OPTIONAL**

  Format: string of 8 characters, 0 to [NW] integers

- **THRESHLD card — OPTIONAL**

  Format: string of 8 characters, 1 real number

- **Type of calculation card — COMPULSORY**

  Format: string of 2 characters, numbers depending on string

**Routine DCBSRW**

`(IPD,NPRIN,KAPPA,Z,E,RG0,RG,RF,MTPX)`

This routine computes the Dirac-Coulomb bound-state orbital radial wavefunction. Eq.(13.5) and (13.5') of Akhiezer and Berestetskii (1957 [2]) are modified to ensure positive slope at the origin for `RG` are used.

The arguments are as follows:

- `IPD` : (input) output stream number

- NPRIN: (input) the (usual) principal quantum number
- KAPPA: (input) the relativistic angular quantum number
- Z : (input) the nuclear charge
- E : (output) the Dirac-Coulomb eigenenergy
- RGO : (output) coefficient of the leading term in the series expansion of the large component near the origin
- RG : (output) $r$ times the large component wavefunction of Akhiezer and Berestetskii
- RF : (output) $r$ times the small component wavefunction of Akhiezer and Berestetskii
- MTPX : (output) maximum tabulation point

Routine(s) called: CGAMMA.

**Routine DEFCOR**
(J)

This routine computes the deferred corrections for orbital J.

Routine(s) called: none

**Routine DEFECT**
(IPD,J,DMNK)

This routine calculates the relativistic quantum defect DMNK, as defined by Johnson and Cheng (1979 [18]), from the one-particle energy ENK.

Routine(s) called: none

**Routine DEL12E**
(IPD,IR1,EAL,EOL,J,JP,DEL1E,DEL2E,DELTAC)

This routine evaluates the variations DEL1E, DEL2E for a rotation in the subspace defined by orbitals J, JP.

Routine(s) called: FCO, GCO, RINTI, SLATER.

**Routine DPBDT**
(J)

This routine computes H times the derivative, with respect to the internal grid, of the large and small components of the wavefunction with index J. These are tabulated, respectively, in arrays TA and TB in common block /TATB/.

A thirteen-point Lagrange formula is used for the calculation of derivatives.

Routine(s) called: none

**Routine DRAW**
(IPD,P,SP,Q,SQ,MF)

This routine generates a printer plot. P and Q are radial functions with the maximum tabulation point MF. SP is the factor by which P is to be scaled, SQ is the factor by which Q is to be scaled.

Routine(s) called: none

**Routine DSTRBN**
(IOPEN,NC,JCFN)

This routine calculates the distribution of electrons from a nonrelativistic subshell over relativistic subshells and calculates their statistical weights.

**Routine DUMP**
(IPD,IP2,IC)

This routine writes the MCDF dump file on datastream IP2. If IC=0, the status label 'ITER' is put on the file header; if IC=1, the status label 'CONV' is written instead.

Routine(s) called: none

**Routine EIGEN**
(IPD,J,RESULT)

This function computes an estimate of the energy of orbital J.

Routine(s) called: DPBDT, QUAD.

**Routine ENGOUT**
(IPD,EAV,E,JTOT,IPAR,ILEV,NN,MODE)

This routine prints energy levels, splittings, and energies relative to the lowest in Hartrees, Kaysers, and eV, using the reduced mass corrected value for the Rydberg. If MODE is 0, only the eigenenergies are printed. If MODE is 1, the eigenenergies and separations are printed. If MODE is 2, the eigenenergies and energies relative to level 1 are printed. If MODE is 3, the eigenenergies, separations, and energies relative to level 1 are printed.

Routine(s) called: none

**Routine ES**
(F,S2F,S3F)

This routine evaluates the sum of the series

$$S_k(f) = \sum_{n=0}^{\infty} (-1)^n \, \frac{\exp(nf)}{n^k} \tag{4.7}$$

for $k = 2$, 3 to machine precision. This is a utility routine, called by routines NUCPOT and NCHARG.

Routine(s) called: none

**Routine ESTIM**
(J)

This routine implements part 1 of algorithm 7.1 of Froese Fischer (1986 [7]) p.320-321.

Routine(s) called: none

**Routine FCO—function**
`(IPD,K,IR,IA,IB)`

This routine evaluates the coefficient

$$f_r^k(a,b) \tag{4.8}$$

Here `K` ($k$) is the multipolarity, `IR` ($r$) is the sequence number of the configuration and `IA` ($a$) and `IB` ($b$) are orbital sequence numbers. See Grant et al (1980 [15]) eq.(6).

Routine(s) called: CLRX.

Option 14 set (ITC-FCO) — Print out of angular coefficient $f_r^k(a,b)$.

**Routine FIXJ**
(JA1,JA2,KA,IS,KS,NS,KJ23)

This routine sets up the arrays `J1`, `J2`, `J3` required by the recoupling package NJGRAF.

Routine(s) called: none

**Routine FUNK—function**
`(IPD,X,N)`

This function evaluates the $K_N(X)$ functions using the analytic functions defined in tables 1 and 3 of Fullerton and Rinker (1976 [8]).

Routines called: none

**Routine FUNL—function**
`(IPD,X,K)`

This function evaluates the $L_K(X)$ functions using the analytic functions defined in table 5 and eq.(20) and (21) of Fullerton and Rinker (1976 [8]).

Routine(s) called: none

**Routine FZALF**
`(IPD,N,KAPPA,Z,VALUE)`

An estimate `VALUE` of the function $F(Z\alpha)$ is computed here.

Routine(s) called: KLAMAQ, MOHR.

**Routine GCO—function**
`(IPD,K,IR,IA,IB)`

This routine evaluates the coefficient

$$G_r^k(a,b) \tag{4.9}$$

Here `K` ($k$) is the multipolarity, `IR` ($r$) is the sequence number of the configuration, and `IA` ($a$) and `IB` ($b$) are orbital sequence numbers. See Grant et al (1980 [15]) eq.(7).

Routine(s) called: CLRX.

Option 14 set (ITC-GCO) — Print of coefficient $G_r^k(a,b)$.

**Routine HEADER**
(IPD)

Routine(s) called: none

**Routine HEADFL**
(NUNIT,GTYPE,GSTAT)

This routine writes standard GRASP file headers. Two records are written.

```
CHARACTER*72 IHED    ... title of run
CHARACTER*8  ITIME   ... time
CHARACTER*8  IDATE   ... date
CHARACTER*4  GTYPE   ... is the GRASP file type.
                         GTYPE = 'BENA'
                         GTYPE = 'MCBP'
                         GTYPE = 'MCDF'
                         GTYPE = 'MCP '
                         GTYPE = 'MCT '
                         GTYPE = 'ORBS'
                         GTYPE = 'OSCL'
CHARACTER*4  GSTAT   ... is the GRASP file status.
                         GSTAT = '    '
                         GSTAT = 'CONV'
                         GSTAT = 'CSF '
                         GSTAT = 'INT '

WRITE (NUNIT) IHED,ITIME,IDATE,GTYPE,GSTAT

INTEGER NW     ... number of orbitals
INTEGER NCF    ... number of CSFs

WRITE (NUNIT) NW,NCF
```

Routine(s) called: none

**Routine HOVLAP**
(IPD,P,Q,MTPO,NP,KAPPA,Z,RESULT)

This routine computes the overlap of the orbital tabulated in the arrays P and Q with maximum tabulation point MTPO with a hydrogenic orbital with parameters NP, KAPPA, Z.

Routine(s) called: DCBSRW, QUAD.

**Routine IMPROV**
(IPD,IR1,CI,EAL,EOL,J)

This routine directs the present implementation of algorithms 5.2 and 5.3 of Froese Fischer (1986 [7]).

Arguments:

J: (input) the index of an orbital.

Routine(s) called: DACON, DAMPCK, DAMPOR, LAGCON, MATRIX, NEWCO, ORTHOR, ORTHSC, QUAD, ROTATE, SETCOF, SETLAG, SOLVE, SXCPOT, TIMER, XPOT, YPOT

**Routine IN**
`(IPD,IORB,JP,P,Q,MTP)`

This routine computes the solution of an inhomogeneous pair of radial Dirac equations in the tail region. A simple extension of the method of Froese Fischer (1963 [5]) is used. The equations are treated as a boundary value problem, with the value of P(R) given at the inner boundary and required to be sufficiently small at the outer boundary. The location of this second boundary is determined in the course of the calculation. The same finite difference equations are used as in the outward integration.

Arguments:

- `IPD` : output stream number `IORB` : (input) index of orbital
- `JP` : (input) tabulation point at which the outward integration terminated
- `P`, `Q` : (input and output) arrays containing, respectively, the large and small components of the solution
- `MTP` : (output) maximum tabulation point of functions `P`, `Q`

When written in matrix form the system of linear equations is $Mw = v$, where $v$ is a vector consisting of the elements `Q(JP)`, `P(JP+1)`, `Q(JP+1)`, $\cdots$. The matrix $M$ is of band type, with 5 elements in each row. $M$ is expressed as a product of two triangular matrices $L$ and $U$ (the $L/U$ decomposition, carried out by the Crout procedure), variable elements of $L$ being stored in the arrays `TC`, `TD` and `TE`, and variable elements of $U$ in `TH`, `TI` and `TJ`. The solution vector $w$ is then obtained by solving two triangular systems, $Lz = v$ and $Uw = z$.

Routine(s) called: none

**Routine INIT**
`(IPD,IR2,IR3,CI,EAL,EOL)`

This routine initialises the MCDF problem.

Routine(s) called: DCBSRW, INTRPQ, ORTHSC, POSNFL, TFPOT, TFWAVE

**Routine INTERP**
`(IPD,XARR,YARR,NARR,XVAL,YVAL,ACCY)`

This routine returns `YVAL` given a value `XVAL` by interpolating using a pair of arrays `XARR(1:NARR)`, `YARR(1:NARR)`, that tabulate a function. `ACCY` is the desired accuracy of the estimate: a warning message is issued if this is not achieved. A warning is also issued

when the routine is extrapolating. Aitken's algorithm is used. See, for instance, Hildebrand (1974 [17]).

**Routine INTRPQ**
`(IPD,PA,QA,MA,RA,J)`

This routine interpolates the arrays `PA(1:MA)`, `QA(1:MA)`, tabulated on grid `RA(1:MA)` into the common arrays `PF(1:MF(J),J)`, `QF(1:MF(J),J)`. (Aitken's algorithm is used. See Hildebrand 1974.)

A check is then made on the norm of the orbital and the orbital is renormalised if necessary.

Routine(s) called: RINT

**Routine IROW1—function**
`(IPD,NELC,KSI)`

This routine locates the row position of configuration $j^n$ in table `NTAB`.

Routine(s) called: none

**Routine ISPARM**
`(IPD,IR1)`

The isotope-shift parameters

$$\Delta E/\Delta C \qquad \text{and} \qquad \Delta E/\Delta A \tag{4.10}$$

for the appropriate energy levels are computed in this routine.

Routine(s) called: NUCPOT, QUAD.

**Routine ITRIG—function**
`(I1,I2,I3)`

The triangular delta. Input: values of $2J+1$; output: 1, if $J$s form a triangle, 0, otherwise.

Routine(s) called: none

**Routine JACOBI**
`(IPD,EOL)`

This routine determines the eigenvalues and eigenvectors of the hamiltonian matrix using the method of Jacobi. Two references are appropriate: Rutishauser (1966 [26]) and Press et al (1986 [23]) sec. 11.1.

Only the upper triangle of `EMT` (in common block /HMAT/) need be given. This is destroyed in the process of diagonalisation. The eigenvalues are returned in `SA` (in common block /HMAT/), the eigenvectors in `DRS` (in common block /SEMI/), both ordered appropriately for the type of calculation as specified by the logical input variable `EOL`.

**Routine JJCSF**
`(IPD,NST,NCF,JMAN,JF,IPAR)`

This routine sets up CSF in $jj$ coupling. It uses a table of allowed angular momenta, seniority, etc., for each occupation number for each subshell to set up CSF in which the angular momenta are successively coupled.

Description of the parameter list:

- NST, NCF : first and last CSF for given configuration
- JMAN : index of current nonrelativistic configuration
- JF : total angular momentum $2J + 1$ for configuration
- IPAR : parity of configuration

Output:

The orbital occupation numbers are stored in array IQ (common block /ORB2/); the total angular momenta, seniority and any other quantum numbers necessary for each subshell are stored in array JQS, and a flag to indicate whether the subshell is full, empty or open (values of $+1$, $-1$ and 0 respectively) is stored in array ICHOP. The angular momenta arising from the coupling of the subshells are stored in array JCUP. These arrays are in common block /STAT/. The total angular momentum and parity of the CSF are stored in arrays ITJPO and ISPAR respectively, in common block /SYM/. The subshell and coupled angular momenta are also stored in arrays JJSUB1, JJSUB2, JJPCUP and JJCUP in common blocks /CUP2/ and /CUP3/.

Note: the routine will handle at most 4 subshells. However, it has been written so that the number of subshells can be easily increased, by copying and changing certain marked sections of the code.

Routine(s) called: none

**Routine JJJLSJ**
(IPD,IBUG3,NST,NCSF,NJJ,JMAN)

This routine transforms subshells from $jj$ to $LS$ coupling.

Description of the parameter list:

- NST, NCSF : first and last CSF for given configuration
- NJJ : number of inner subshells to remain in $jj$ coupling
- JMAN : index of current nonrelativistic configuration

Output:

The transformation matrix is stored in array TC in common block /CUP0/. The subshell angular momenta are stored in array NVSL in common block /NRD6/.

Routine(s) called: LSTERM, SSTC, LTRG, ORTHOG.

**Routine JJRECP**
(IPD,NST,NCSF,JMAN)

This routine recouples subshell $J$ values from the order

$$\cdots(((\cdots)J_{1,\ldots,k-1}J_k)J'_kJ_k)J_{1,\ldots,k}\cdots \tag{4.11}$$

to the order

$$\cdots((\cdots)J_{1,\ldots,k-1}(J_kJ_k)J_k)J_{1,\ldots,k}\cdots \tag{4.12}$$

Description of the parameter list

- `NST`,`NCSF` : first and last CSF for given configuration
- `JMAN` : index of current nonrelativistic configuration

Output

The transformation matrix is stored in array `TC` in common block /CUP0/. The subshell angular momenta are stored in array `JJSCUP` in common block /CUP3/.

Routine(s) called: LTRG, DRACAH, ORTHOG.

**Routine JLABEL—block data**

This block data sets up strings giving $j$-values and parity signs. The string for angular momentum $j$ is that in the array location `2J+1`. If parity is stored as $-1$ for odd parity and 1 for even parity, the appropriate array element is in location `(3+PARITY)/2`.

**Routine JLKSJ**
`(IPD,NST,NCF,NOPEN,NJJ,KSUB)`

This routine recouples the last subshell from the scheme

$$((\cdots)J_{1,\ldots,n-1}(L_nS_n)J_n)J \tag{4.13}$$

to the scheme

$$(((\cdots)J_{1,\ldots,n-1}L_n)KS_n)J \tag{4.14}$$

Description of the parameter list

- `NST`,`NCF` : first and last CSF for given configuration
- `NOPEN` : number of open shells in configuration
- `NJJ` : number of inner subshells to remain in $jj$ coupling
- `KSUB` : array containing list of (open) subshell indices in required order of coupling (innermost first)

Output

The transformation matrix is stored in array `TC` in common block /CUP0/. The `K` values are stored in array `NVSL` in common block /NRD6/.

Routine(s) called: DRACAH, LTRG, ORTHOG.

**Routine JRECUP**
`(IPD,IBUG3,NST,NCSF,JMAN,KSUB,LPAIR)`

This routine recouples subshells from the standard order to the required order.

Description of the parameter list:

- `NST`, `NCSF` : first and last CSF for given configuration
- `JMAN` : index of current nonrelativistic configuration
- `KSUB` : array containing list of (open) subshell indices in required order of coupling (innermost first)
- `LPAIR` : logical array indicating whether subshell is first of a pair of subshells which are to be coupled together before being coupled to the last resultant

Output:

The transformation matrix is stored in array `TC` in common block /CUP0/. The subshell angular momenta are stored in array `JJCUP` in common block /CUP3/.

Note: the routine will handle at most four (4) subshells. However, it has been written so that the number of subshells can be easily increased, by copying and changing certain marked sections of the code.

Routine(s) called: NJGRAF, GENSUM, ORTHOG

**Routine KLAMAQ**
`(IPD,N,KAPPA,Z,FZALFA)`

The function $F(Z\alpha)$ is estimated here. We use the series expansion given by eq.(1) and eq.(2) and the table of Bethe logarithms in Klarsfeld and Maquet (1973 [19]). The vacuum polarisation contribution in eq.(2) is omitted.

**Routine KNJ**
```
(JD6C,JD7C,JD8C,JD9C,JDWC,JD6,JD7,JD8,JD9,JDW,JDDEL,
LDDEL,MDP,JD6P,JD7P,JD8P,JD9P,JDWORD,NDLSUM,
NDBJ,NDB6J,KD6CP,KD7CP,KD8CP,KD9CP,JDSUM4,JDSUM5,
JDSUM6,INVD6J)
```

This routine stores data for future calls to GENSUM.

**Routine LAGCON**
`(J)`

This routine includes the Lagrange multiplier contribution in the 'exchange' term.

Routine(s) called: none

**Routine LLLSSS**
`(IPD,IBUG3,NST,NCF,NJJ,NOPEN,NSUB,KSUB,LPAIR)`

This routine recouples from intermediate to $LS$ coupling.

Description of the parameter list:

- NST, NCF : first and last CSF for given configuration
- NJJ : number of inner subshells to remain in $jj$ coupling
- NOPEN : number of open shells in configuration
- NSUB : number of subshells to be transformed
- KSUB : array containing list of (open) subshell indices in required order of coupling (innermost first)
- LPAIR : logical array indicating whether subshell is first of a pair of subshells which are to be coupled together before being coupled to the last resultant

Output:

The transformation matrix is stored in array TC in common block /CUP0/. The subshell angular momenta are stored in array NVSL in common block /NRD6/.

Note: the routine will handle at most four (4) subshells. However, it has been written so that the number of subshells can be easily increased, by copying and changing certain marked sections of the code.

Routine(s) called: NJGRAF, GENSUM, LTRG, ORTHOG.

**Routine LNGTH—function**
(STRING)

**Routine LOAD**
(IPD,IDT,NREC)

This routine reads the MCDF dump file.

Routine(s) called: DATOUT, HEADER, SETQIC.

**Routine LSDATA—block data**

**Routine LSTERM**
(IPD,JMAN)

This routine sets up $LS$ terms for each unfilled subshell, storing $V$, $S$ and $L$ values in the array NLSVT. Limitations on configurations are (1) $q < 3$ and (2) $l < 3$, $q > 3$, where $q$ is the reduced occupation number, i.e. $4l + 2 - q$ if $q > 2l + 1$, $q$ otherwise, and $l$ is the subshell orbital angular momentum.

Routine(s) called: none

**Routine LTAB**
(IPD,IS,NQS,KS,IROWS)

This routine locates rows of possible parents of active shell states for accessing NTAB. It is assumed that empty shells have been eliminated from consideration by routine RKCO.

Routine(s) called: none

**Routine LTRG—function**
`(J1,J2,J3)`

This function tests the triangle condition on three angular momenta. It returns the value .TRUE. if the condition is satisfied, .FALSE. otherwise. `J1`, `J2` and `J3` are the $2J+1$ values of the angular momenta.

**Routine MANOUT**
`(IPD)`

This routine prints CSF data in a form that can be read by routine DATR in MCDF program.

**Routine MATOUT**
`(IPD,HAMIL,ICOL,NR,NC,NRX,NCX,MODE)`

This routine prints matrices:

`MODE` = 1 symmetric
`MODE` = 2 unsymmetric
`MODE` = 3 matrix of eigenvectors

Routine(s) called: none

**Routine MATRIX**
`(IPD,IR1,CI,EAL,EOL)`

This routine calls routines to form the hamiltonian matrix and to diagonalise it. The total angular momenta of the ASF are found and the eigenvectors are chosen so that the sign of the largest element is positive.

Routine(s) called: ENGOUT, JACOBI, MATOUT, SETHAM.

Option 9 set (ITC-MATRIX) — Print hamiltonian matrix.

Option 10 set (ITC-MATRIX) — Print all eigenvalues and eigenvectors for CI case.

**Routine MAXARR**
`(J)`

This routine finds the least self-consistent orbital.

Routine(s) called: none

**Routine MCBP**
`(IPD,IBUG1,IBUG2,IBUG3,NOUT,NIN)`

This is the main routine of the MCBP package. This package determines the values of and all indices of the angular coefficients

$$V_{rs}^{kt}(abcd) \tag{4.15}$$

required for the evaluation of the transverse photon interaction.

Routine(s) called: CHEKFL, CLOSFL, HEADFL, OPENFL, TMSOUT, RKCO, RESTMB.

**Routine MCDF**
`(IPD,IR1,IR2,IP2,IR3,IP3)`

This routine solves the MCDF problem. The algorithm of Froese Fischer (1986 [7]) is followed to the extent appropriate for each of the CI/(E)AL/(E)OL calculations.

Routine(s) called: CPPOT, DUMP, INIT, MATOUT, MATRIX, NEWBAS, NRCWT, NUCPOT, ORBOUT, PROP, PRWF, RADGRD, SCF, SETIOM, WEIGHT

**Routine MCP**
`(IPD,IBUG1,IBUG2,IBUG3,IBUG4,IBUG6,NOUT,NSORT,NIN)`

This is the main routine of the MCP package. This package determines the values of and all indices of the angular coefficients

$$T_{rs}(ab) \qquad \text{and} \qquad V^k_{rs}(abcd) \tag{4.16}$$

$k$ is the multipolarity of a two-particle Coulomb integral. $a$, $b$, $c$ and $d$ are orbital sequence numbers. $r$ and $s$ are configuration state function indices.

Two output files, with unit numbers `NOUT` and `NSORT` are generated. We refer to them, respectively, as the unsorted and sorted MCP output files. The program will restart from an incomplete unsorted MCP output file. The unit number of the latter is `NIN`.

Routine(s) called: CHEKFL, CLOSFL, HEADFL, OPENFL, REMAIN, RKCO, TNSRJJ, TRSORT.

This is a modification of MCP75 by Grant (1976 [13]). The correction deck described in Grant (1978 [14]) has been incorporated.

**Routine MCT**
`(IPD,IBUG1,IBUG3,IBUG4,IBUG6,NOUT,NSORT,NIN)`

This is the main routine of the MCT package. This package determines the values of and all indices of the angular coefficients

$$D^k_{ab}(rs) \tag{4.17}$$

for tensor operators of ranks `KA(I)` and parities `IOPAR(I)`, `I=1,..,NKA`. $a$ and $b$ are orbital sequence numbers. $r$ and $s$ are configuration state function sequence numbers.

Routine(s) called: CHEKFL, CLOSFL, HEADFL, OPENFL, REMAIN, TNSRJJ, TRSORT.

**Routine MCTIN**
`(IPD,NFILE,IOPAR,JKP)`

This routine operates in two modes decided by `NKP`, which is passed in common block /OSC6/. When `NKP` is zero, the routine checks the MCT file and sets `NKP` to the number of rank/parity combinations in the MCT file. This is an initialisation call. When `NKP` has some positive value, this routine loads information about the coefficients with parity and rank specified by `KP(JKP)` into the arrays `ISLDR` and `XSLDR`. The stream `NFILE` contains the MCT coefficients. `IOPAR` is the parity ($\pm 1$) and is determined from the sign of `KP(JKP)`.

Routine(s) called: none

**Routine MODJ23**

This routine restores common block /COUPLE/ from saved values for exchange case.

Routine(s) called: none

**Routine MOHR**
`(IPD,N,KAPPA,Z,FZALFA)`

The function $F(Z\alpha)$ for the 1s, 2s, 2p−, 2p symmetries is computed here. A value is obtained by interpolating in, or extrapolating from, the table due to Mohr (1983 [22]).

Routine(s) called: INTERP.

**Routine MUMDAD**
`(IPD,IBUG2,IS,KAPS,X)`

This routine evaluates the product of 4 CFPs.

Routine(s) called: CFP.

**Routine NCHARG**

This routine evaluates the nuclear charge density, and stores it in the common array `ZZ`.

Routine(s) called: ES.

**Routine NEWBAS**
`(IPD)`

This routine transforms eigenvectors from $jj$ to $LS$ CSF basis.

**Routine NEWCO**
`(IPD,EOL)`

This routine computes the configuration mixing coefficients and generalised occupation numbers for OL/EOL calculations, given the eigenvectors of the hamiltonian.

Routine(s) called: none

**Routine NEWE**
`(J,SGN,NPRIME,MX,DELEPS,FAIL,INV)`

This routine implements part 2 of algorithm 7.1 in Froese Fischer (1986 [7]). The present code actually implements the version used in the program MCHF where differences occur.

Routine(s) called: OUTBND

**Routine NRCWT**
`(IPD)`

Print the weights of the largest five nonrelativistic configuration contributors to each ASF.

**Routine NUCPOT**
`(IPD,VERBOS)`

This routine evaluates the nuclear potential for point and Fermi models.

Routine(s) called: DRAW, ES.

Option 2 set (ITC-NUCPOT) — Print nuclear potential when called from routine MCDF.

**Routine OCON—function**
`(IA1,IB1,IA2,IB2)`

This routine evaluates the multiplicative statistical factor. It is assumed that states are ordered so that `IA1`≤`IB1`, `IA2`≤`IB2`.

Routine(s) called: none

**Routine OPENFL**
`(IPD,G2NAME,STAT,NUNIT)`

This routine opens files for unformatted I/O.

`G2NAME` is an input. It is a string describing the file.

`STAT` is the status of the file, one of 'NEW' or 'OLD'.

`NUNIT` is the number of the FORTRAN file unit.

**Routine ORBOUT**
`(IPD,IR3,IP3)`

This routine writes orbital data to the "WRITE ORBITALS" file. If MCDF option 22 is set, the "READ ORBITALS" file is appended to the "WRITE ORBITALS" file before orbital data is appended.

Three records of information are written for each orbital.

- Record 1 : `NP(J),NAK(J),E(J),MFJ`
- Record 2 : `PZ(J),(PF(I,J),I=1,MFJ),(QF(I,J),I=1,MFJ)`
- Record 3 : `(R(I),I=1,MFJ)`

Routine(s) called: POSNFL.

**Routine ORDER**
`(IPD)`

This routine calls the diagonalisation routines and matches the eigenvalues and eigenvectors of the higher order matrix to those of the zero order matrix.

Routine(s) called: JACOBI, MATOUT

**Routine ORTHOG**
`(IPD,NST,NCF,CALLER)`

This routine checks the orthonormality of the transformation.

Routine(s) called: none.

**Routine ORTHOR**
`(IPD,J,INV)`

This routine Schmidt orthogonalises orbital `J` to all orbitals which have better self-consistency. Note that fixed orbitals have the best self-consistency.

Routine(s) called: COUNTX, RINT.

**Routine ORTHSC**
`(IPD,IMODE)`

This routine performs Schmidt orthogonalisation. If `IMODE` is 0, all fixed orbitals are first orthogonalised in the order defined by the card input. The procedure for `IMODE` = 1 is then followed. If `IMODE` is 1, all orbitals are first orthogonalised to the fixed ones, then from the first to last as defined by the card input. The program prints overlap integrals.

Routine(s) called: COUNTX, RINT.

**Routine OSCL**
`(IPD,IMCDF,IMCT,ITRPR)`

This routine controls the main sequence of routine calls for the calculation of data for transitions between multiconfiguration Dirac-Fock energy levels.

Routine(s) called: BESSJ, CHEKFL, CLOSFL, CSFM, ENGOUT, HEADFL, LOAD, OPENFL, MATOUT, MCTIN, PRINTA

Select energies and eigenvalues for calculation. Default — Use best available.

Option 1 (LTC-OSCL) — Use Coulomb results only.
Option 2 (LTC-OSCL) — Use Coulomb+Breit results only.
Option 3 (LTC-OSCL) — Use Coulomb results as well as best results.
Option 4 (LTC-OSCL) — Use Coulomb+Breit results as well as best results.

Option 5 (LTC-OSCL) — Set `NLP` = {`NLP`—:70}−7. Default — `NLP` = `NLP`—:70−8.

Set up list of levels for calculation of oscillator strengths.

Option 6 (LTC-OSCL) — Sort list into increasing order of energy.

Set up units for printing transition energy. Default — Hartrees.

Option 7 (LTC-OSCL) — Print transition energy in Angstroms.
Option 8 (LTC-OSCL) — Print transition energy in Kaysers.
Option 9 (LTC-OSCL) — Print transition energy in eV.
Option 10 (LTC-OSCL) — Print transition energy in Hz.

Option 19 (LTC-OSCL) — Print eigenvalues and eigenvectors.

**Routine OUT**
`(J,JP,P,Q)`

This routine carries out the step-by-step outward integration of a pair of inhomogeneous Dirac radial equations.

Arguments:

- J: (input) orbital index of function to be computed
- JP: (input) the join point; the outward integration stops at this tabulation index
- P,Q: (input and output) on input, elements 1 to 3 of both arrays must be tabulated; on output, the arrays are tabulated up to point JP

Routine(s) called: none

**Routine OUTBND—function**
`(ETRY)`

This routine determines whether the trial eigenvalue `ETRY` is within the bounds (`EPSMIN`,`EPSMAX`)

Routine(s) called: none

**Routine POLFAX**
`(I,J)`

This routine modifies the Bessel function arrays to include the core-polarisation contribution for electric dipole transitions.

Routine(s) called: none

**Routine POSNFL**
`(NUNIT)`

This routine positions standard GRASP files at the end of the header.

Routine(s) called: none

**Routine PRINTA**
`(IPD,ASFA,ASFB,I,J,OMEGA,FACTOR,LINES,ITRPR)`

This routine prints the basic oscillator strength information for transitions between level `I` and level `J`.

Routine(s) called: none

**Routine PROP**
`(IPD)`

This routine calculates and/or prints orbital and level properties.

Routine(s) called: DEFECT, ENGOUT, MATOUT, RINT, ZEFR, ZEFE

**Routine PRWF**
`(IPD,J)`

This routine prints wavefunctions. There are two modes:

- `J`$> 0$ — used as a debug option in SOLVE, wavefunctions for orbital `J` are printed
- `J`$= 0$ — a printout of the grid and all wavefunctions is made

Routine(s) called: DRAW.

**Routine QED**
`(IPD,IMCP)`

This routine estimates the QED corrections to the energy levels due to self-energy and vacuum polarisation.

Routine(s) called: FZALF, HOVLAP, NCHARG, QUAD, SCREEN, VACPOL.

**Routine QUAD**
`(IPD,RESULT)`

The argument `RESULT` is an approximation to the integral of $f(r)$ from zero to infinity, where the values of `RP(I)*F(R(I))` are tabulated in the array `TA(I)`. The integral in the interval zero to `R(2)` is computed by use of an analytical fit

$$f(r) = A\, r^{\sigma} \tag{4.18}$$

A five-point closed Newton-Cotes formula (cf. Hildebrand (1974 [17]) p.93) is used to compute the integral in the interval `R(2:MTP)`. The contribution from the tail of the function beyond the last tabular point (`MTP`) is assumed to be negligible. The method uses `MTP+3` tabulation points. Array `TA` should therefore be dimensioned to at least `N+4`.

Routine(s) called: none

**Routine RADGRD**
`(IPD)`

This routine sets up the radial grid `R` and the associated arrays `RP` and `RPOR` in the common block /GRID/. Different grids are generated depending on whether or not `HP` is zero.

Exponential grid:

R(I) = RNT*(EXP((I-1)*H)-1) for I = 1, ..., N

Asymptotically-linear exponential grid:

LN(R(I)/RNT+1)+(H/HP)*R(I) = (I-1)*H for I = 1, ..., N

Routine(s) called: DRAW, SETQIC

Option 1 set (ITC-RADGRD) — Print radial grid.

**Routine RBAR—function**
`(N,K,Z)`

This routine computes the expectation value of $r$ for a Coulomb function with quantum numbers $n$, $\kappa$ for nuclear charge $Z$.

**Routine REMAIN**
`(IPD,NIN,NOUT,JASTRT,JBSTRT)`

This routine determines the CSF pair with which the MCP package should start computation. All file checking is assumed to have been carried out before entry into this code.

**Routine REORDR**
(NST,NOPEN)

This routine reorders the CSFs on the basis of: (1) total $J$; (2) occupation number; (3) subshell $j$ value and quantum numbers; (4) coupled $J$ values. A linked list sorting method is used.

Routine(s) called: none

**Routine RESTMB**
(IPD,NIN,NOUT,JASTRT,JBSTRT)

This routine determines where to start or restart the calculation of MCBP coefficients.

Routine(s) called: none

**Routine RINT**
(IPD,JA,JB,JK,RESULT)

The value of this function is an approximation to:

$$\int_0^\infty \left(r^k \left(P_i P_j + Q_i Q_j\right)\right) \tag{4.19}$$

Routine(s) called: QUAD.

**Routine RINTI**
(IPD,JA,JB,MODE,RESULT)

The value of this function is the one-electron integral $I(j,k)$ for orbitals J, K. The analytical expression for this is given by eq.(9) in Grant et al (1980 [15]).

Routine(s) called: DPBDT, QUAD.

Option 12 set (ITC-RINTI) — Print one-electron integral.

**Routine RKCO**
(IPD,IBUG1,IBUG2,IBUG3,JA,JB,COR,CORD,INCOR)

For configurations JA, JB, this routine Analyses the tables of quantum numbers set in the common blocks /M0/, /M1/, /M2/, /M3/ to determine all possible sets of interacting orbitals which give a non-vanishing Coulomb matrix element, and initiates the calculation of coefficients.

The following conventions are in force: (1) labels 1, 2 refer to left, right sides of matrix element respectively. (2) pointers JA1, JB1, JA2, JB2 point to the JLIST array of active orbitals; IA1, IB1, IA2, IB2 point to the complete list of orbitals.

Routine(s) called: COR, CORD, SETUP, VIJOUT.

- 1.0 Analyse peel shell interactions
- 1.1 Analyse electron distribution in peel. (The full procedure is needed only if the number of peel orbitals NPEEL$\geq$2)

  Find differences in occupations, NDQ, for each peel orbital in turn and use to set up labels of active orbitals maintaining the convention JA1$\leq$JB1, JA2$\leq$JB2.

- 1.2 Calculate coefficients for all possible sets of active shells.

  There are 4 cases, depending on the value of IDQ, the sum of the absolute differences NDQ:

  1.2.1 IDQ>4 : matrix element null

  1.2.3 IDQ=2 : one orbital fixed each side include all possible spectators.

  Also IDQ=0 for a matrix element off-diagonal in coupling only. Must sum over all pairs of orbitals excluding core-core terms.

  This section calculates the terms arising from active electrons which are in closed shells.

  1.2.4 IDQ=0 - diagonal case. Include all pairs with JA1=JA2, JB1=JB2.

- 2.0 The diagonal case. Deal with contributions from core orbitals if INCOR=1.
- 2.1 Calculate contribution from core/core terms
- 2.2 Calculate contribution from peel/core terms
- 3.0 Diagnostic print - NW<1

**Routine RKINT**
(IPD,RAC,IA,IC,RBD,IB,ID,K,IW,RESULT)

This routine evaluates the Breit interaction integrals. If IW=0, it calculates $U(r_1, r_2)$ integral. If IW = 1,2, it calculates $\bar{R}(k; ac|bd; \omega)$ with $\omega = \omega_{ac}$ if IC = 1, $\omega = \omega_{bd}$ if IC = 2.

Routine(s) called: QUAD, ZKF

**Routine ROTATE**
(IPD,IR1,EAL,EOL,FIRST)

This routine rotates the orbital basis. Each pair of orbitals that is constrained by orthogonality is treated by turn. The method used is due to Froese Fischer (1986 [7]).

Routine(s) called: COUNT, DEL12E, NEWCO, RINT.

Option 27 set (ITC-ROTATE) — Perform rotation analysis.

**Routine SBSTEP**
(IORB,NSTRT,NEND,P,Q)

This routine continues the solution of the homogeneous Dirac radial equation from tabulation point NSTRT to tabulation point NEND. The algorithm of Sienkiewicz and Baylis (1987 [27]) p.5155, is used.

Routine(s) called: none

**Routine SCF**
(IPD,IR1,IR2,IP2,CI,EAL,EOL)

This routine performs the SCF iterations. The procedure is essentially algorithm 5.1 of Froese Fischer (1986 [7]).

Routine(s) called: DUMP, IMPROV, MATRIX, MAXARR, NEWCO, ORTHSC, PRWF, ROTATE, SETLAG, TIMER

**Routine SCREEN—function**
`(IPD,J)`

This routine estimates the screening (or shielding) for orbital `J`. The set of rules used by Froese Fischer in her program MCHF are used.

Routine(s) called: none

**Routine SETCOF**
`(IPD,IR1,J)`

This routine sets up the coefficients and orbital pointers for the direct and exchange potentials for orbital `J`. It also sets up the coefficients and pointers for the inhomogeneous terms arising from off-diagonal `I(a,b)` integrals.

Routine(s) called: none

**Routine SETCON**
`(IPD)`

This routine sets the values of the fundamental and derived physical constants, and other useful constants.

**Routine SETHAM**
`(IPD,IR1)`

This routine computes the hamiltonian matrix and determines the average energy.

Routine(s) called: CLRX, FCO, GCO, RINTI, SLATER.

**Routine SETIOM**
`(IPD,IR1,IR2,IP2,IR3,IP3)`

This routine sets up the I/O datastreams for the MCDF (and SCF) modules. Some checking is performed.

Routine(s) called: CHEKFL, OPENFL

**Routine SETJ**
`(IS,JS,KS,NS,KJ23)`

This routine sets the tables required by the recoupling coefficient package NJGRAF. This routine loads the common block /COUPLE/ with parameters for the first call of NJGRAF involving direct integrals. Subsequent exchange calls of NJGRAF must be preceded by a call of MODJ23 to restore these arrays to their correct initial state.

Routine(s) called: none

**Routine SETLAG**
`(IPD,IR1,FIRST)`

If argument FIRST is .TRUE. this routine sets up the data structure pertaining to the Lagrange multipliers. If FIRST is .FALSE. it determines new estimates for the multipliers.

Routine(s) called: DACON, QUAD, RINTI, SETCOF, XPOT, YPOT.

**Routine SETMC**
(IPD)

This routine performs machine-dependent setup.

**Routine SETPOT**
(IPD,J,JP)

This routine sets up the arrays TF and TG for use by the routines IN and OUT.

Arguments:

- J : (input) index of orbital
- JP : (output) join point; point where TG changes sign

Routine(s) called: none

**Routine SETQIC**

This routine sets up the coefficients for routines DPBDT, QUAD, RINTI, START, YZK, ZKF.

Routine(s) called: none

**Routine SETUP**
(IPD,JA,JB)

This generates the arrays defining the quantum numbers of the states involved in the matrix element linking configurations labelled by JA, JB.

Routine(s) called: none

**Routine SETXUV**
(J)

This routine sets up the arrays XU and XV, for use by the routines IN and OUT.

Routine(s) called: none

**Routine SETXV**
(J)

This routine sets up the inhomogeneous terms for the variation equations.

Routine(s) called: none

**Routine SETXZ**
(J)

This routine sets the inhomogeneous terms to zero.

Routine(s) called: none

**Routine SKINT**
`(IPD,RAC,IA,IC,RBD,IB,ID,K,IW,RESULT)`

This routine evaluates Breit interaction integrals:

$$S^{(k)}(a,c;b,d;\omega) \tag{4.20}$$

where $\omega = \omega_{ac}$ if `IW` $= 1$, and $\omega = \omega_{bd}$ if `IW` $= 2$.

Routine(s) called: QUAD, ZKF

**Routine SKRC**
`(IS,KAPS,KS,KD1,KD2,KE1,KE2)`

This routine determines the range of the tensor rank $k$ for Coulomb integral.

Routine(s) called: none

**Routine SLATER**
`(IPD,JA,JB,JC,JD,JK,RESULT)`

The value of this function is the Slater integral as normally defined in terms of the four sets of quantum numbers $a, b, c, d$.

Routine(s) called: QUAD, YZK.

Option 11 set (ITC-SLATER) — Print Slater integral.

**Routine SNRC**
`(IS,KAPS,KS,ND1,ND2,NE1,NE2,IBRD,IBRE)`

This routine determines the range of tensor rank `NU` for direct/exchange terms, and classifies the types of radial integral.

Input variables:

- `IS` : orbital labels
- `KAPS` : values of $2\kappa$
- `KS` : values of $2j + 1$

Outputs:

- `ND1/NE1` : lowest `NU` value for direct/exchange types
- `ND2/NE2` : corresponding number of contributing `NU` values: `NU` = `ND1`, `ND1+2`, $\cdots$, `ND1+2*(ND2-1)` etc
- `IBRD/IBRE` : classify types of radial integrals contributing; negative value implies null contribution

Routine(s) called: none

**Routine SOLVE**
`(IPD,J,FAIL,INV,JP,NNP)`

This routine performs step 2 in algorithm 5.2 and 5.3 of Froese Fischer (1986 [7]). Some minor changes have been made.

Arguments:

- `J` : (input) the serial number of the orbital
- `JP` : (output) the join point
- `FAIL` : (output) if .TRUE., the iterations did not yield an acceptable solution (methods 1 and 2)

Routine(s) called: COUNT, DCBSRW, ESTIM, IN, NEWE, OUT, PRWF, QUAD, SETPOT, SETXUV, SETXV, SETXZ, START

Option 7 set (ITC-SOLVE) — Debug print of iterations.

Option 8 set (ITC-SOLVE) — Debug print of wavefunctions by call to routine PRWF.

**Routine SOLVH**
`(IPD,IORB,FAIL)`

This routine solves the homogeneous Dirac radial equation.

Arguments: IORB : (input) index of orbital FAIL : (output) .TRUE. if solution not obtained

The direct potential is assumed tabulated in the common array `YP`.

Routine(s) called: COUNTX, QUAD, SBSTEP, SETPOT, START, TAIL.

**Routine SPEAK**
`(IPD,IBUG1,IA1,IB1,IA2,IB2,K,X)`

This routine outputs MCP coefficients and integral parameters to file on stream `NOUTX`. Also print these if `IBUG1` = 1.

Routine(s) called: none

**Routine SPME**
`(IPD,I,J,HCOUL,HBAB,HMAG)`

This routine calculates the reduced matrix elements for pair `I`, `J` in either Coulomb/Babushkin gauge or for magnetic case.

These are defined in the Brink and Satchler (1993 [4]) sense - i.e. compatible with Pyper et al (1978 [24]) but not with Grant (1974 [12]).

Routine(s) called: CLRX, POLFAX, QUAD.

Option 12 set (LTC-SPME) — Print integrals.

Option 13 set (LTC-SPME) — Print gauge dependent integrand.

Option 14 set (LTC-SPME) — Calculate the contributions from various terms.

Option 15 set (LTC-SPME) — Full print if option 14 set; abbreviated otherwise.

**Routine SSTC**
`(IPD,IBUG3,`
`NLSUB,NQ,NJ,NL,NS,NV,NQR1,NJ1,NV1,NJ2,NV2,COEFF)`

This routine uses a table look-up to find $JJ - LS$ transformation coefficients for single (nonrelativistic) subshells for s, p and d electrons. For orbitals with higher angular momenta and not more than two electrons, the coefficient is calculated from the relevant $9j$-symbol.

Routine(s) called: NJGRAF, GENSUM, KNJ.

**Routine START**
`(IPD,IORB,ITYPE,P0,P,Q0,Q)`

This routine sets up `P(1:6)`, `Q(1:6)`, required to start the integration for routines OUT and SBSTEP.

Arguments:

- `IORB` : (input) index of the orbital
- `ITYPE` : (input) 1 = homogeneous equation; 2 = inhomogeneous equation; 3 = variational equation
- `P0` : (input) slope parameter
- `P` : (output) `P(1:6)` are tabulated by this routine
- `Q0` : (output) first term in the series expansion of $Q$
- `Q` : (output) `Q(1:6)` are tabulated by this routine

Routine(s) called: none

**Routine SUMMRY**
`(IPD)`

This routine gives a final summary of the contributions to the energy levels from zero-order, Breit, vacuum polarisation and self energy. These are given in Rydbergs, Kaysers and eV if options 27, 28 and 29 are set, respectively, as well as in Hartrees. The energy levels are written out in order of increasing energy.

Routine(s) called: none

**Routine SXCPOT**
`(IPD)`

This routine computes a central potential. Relativistic statistical exchange and correlation contributions are calculated.

There are two possiblities for the nonrelativistic statistical exchange potential. If SEPARM is 1 it is the Gaspar-Kohn-Sham exchange potential. If SEPARM is 3/2 it is the Slater exchange potential.

The relativistic exchange correction factor is given by AH MacDonald and SH Vosko (1979 [20]), eq.(3.14C).

The relativistic local density approximation statistical correlation potential was computed by Ramana and Rajagopal (1981 [25]), table I. The high-density values of VC are a private communication. A high-precision recomputation of this table and its extension to both higher and lower BETA would be most useful.

The relativistic statistical correlation potential is obtained by interpolating among the values in the table of Ramana and Rajagopal.

Routine(s) called: DRAW, YZK

Option 6 set (ITC-SXCPOT) — Print statistical potential.

**Routine TAIL**
(IPD,IORB,P,Q,JP,MTP)

This routine begins the inward integration of the homogeneous Dirac radial equation. With only minor modifications, the series given by Sienkiewicz and Baylis (1987 [27]) p.5155, is used.

Routine(s) called: none

**Routine TALK**
(IPD,IBUG1,NU,IA,IB,IC,ID,ITYPE,COEF)

This routine prints coefficients and integral parameters if IBUG1$>0$ and writes them to file.

Routine(s) called: none

**Routine TERM—block data**

This block data sets up tables of quantum numbers of terms which can be formed by configurations $j^q$. Symmetry of the table for particle/hole configurations is used to compress it.

COMMON /TERMS/NROWS,ITAB(31),JTAB(31),NTAB(327)

A row is defined by a subshell angular momentum and an occupation number.

Each entry ITAB gives the number of terms in a row.

Each entry JTAB gives the (starting location $-1$) of the first triad a row.

Each triad in NTAB is $(v, w, 2J+1)$; here $v$ is the seniority, $w$ resolves any degeneracy in the seniority scheme and $J$ is the total angular momentum.

**Routine TFPOT**
(IPD)

This routine calculates the universal Thomas-Fermi potential.

Routine(s) called: DRAW.

**Routine TFWAVE**
`(IPD,J,MFAIL)`

This routine is used to produce estimates of the wavefunctions by use of the Thomas-Fermi approximation to the direct potential. Routine SOLVH is used to obtain the radial wavefunctions.

Routine(s) called: SOLVH.

**Routine TMSOUT**

This routine prints the table of relativistic subshell quantum numbers set in block data TERMS .

Routine(s) called: none

**Routine TNSRJJ**
`(IPD,IBUG3,IBUG4,IBUG6,`
`KA,IOPAR,JA,JB,IA1,IA2,VSHELL)`

The main program for evaluating the reduced matrix elements of a one particle operator for configurations in $jj$-coupling.

Routine(s) called: CFP, FIXJ, GENSUM, IROW1, ITRIG, NJGRAF, SETUP, VIJOUT.

**Routine TRSORT**
`(IPD,IREAD,IWRITE,GTYPE,IPRINT)`

This routine sorts MCP coefficients into list based on integrals rather than CSF. A tree sort is used to save space, the CSF pair labels and coefficients are not read into arrays until the sorting has been done.

Routine(s) called: HEADFL, POSNFL.

**Routine VAC2**
`(IPD)`

This routine sets up the second-order vacuum polarisation potential using eq.(1) and (4) of Fullerton and Rinker (1976 [8]). The potential is accumulated in array `TB(I),I=1,...,N` which is in common block /TATB/.

Routine(s) called: FUNK, QUAD.

Option 19 set (KTC-VAC2) — Print vacuum polarisation potential.

**Routine VAC4**
`(IPD)`

This routine sets up the fourth-order vacuum polarisation potential using eq.(11) and (12) of Fullerton and Rinker (1976 [8]). The potential is accumulated in array `TC(I),I=1,...,N`. It is added to the second-order vacuum polarisation potential and transferred to array `TB` in common block /TATB/.

Routine(s) called: FUNL, QUAD.

Option 9 set (KTC-VAC4) — Only second-order vacuum polarisation potential is required.

Option 19 set (KTC-VAC4) — Print vacuum polarisation potential.

**Routine VACPOL**
`(IPD)`

This routine controls the setting up of the vacuum polarisation potential for the given nuclear charge distribution at each grid point using the analytic functions defined by Fullerton and Rinker (1976 [8]). The potential is accumulated in array `TB(I),I=1,...,N` which is in common block /TATB/.

Routine(s) called: VAC2, VAC4, VACUSR.

Option 7 set (KTC-VACPOL) — Use user-defined vacuum polarisation potential. This is set in routine VACUSR.

**Routine VACUSR**
`(IPD)`

User-defined vacuum polarisation potential. User must supply code and delete error section. The potential is stored in array `TB` of common block /TATB/.

Option 19 set (KTC-VACUSR) — Print user-defined vacuum polarisation potential.

**Routine VIJOUT**
`(IPD,JA,JB)`

This routine prints out tables of configurational quantum numbers defined by routine SETUP for current matrix element.

Routine(s) called: none

**Routine WEIGHT**
`(IPD)`

This routine prints the weights of the largest five CSF contributors to each ASF. Also it prints weights of configurations.

**Routine WFN—function**
`(R,RCUT,N)`

This routine calculates the cutoff function

$$W_n(r, r_c) = 1 - \exp\left(-(r/r_c)^n\right) \tag{4.21}$$

for use in routines CPPOT and POLFAX.

Routine(s) called: none

**Routine XPOT**
`(IPD,J)`

This routine tabulates the exchange terms (the first terms on the right-hand sides of eq.(14), Grant et al (1980 [15]) for orbital `J`. The exchange terms are stored in the common arrays `XP` and `XQ`.

Routine(s) called: DRAW, YZK

**Routine YPOT**
`(IPD,J)`

This routine tabulates the potential function $Y(r)$ (eq.(14) in Grant et al (1980 [15])) for orbital `J`. The function is tabulated in the common array `YP`.

Routine(s) called: DRAW, YZK

Option 6 set (ITC-YPOT) — Print out direct potential.

Option 15 set (ITC-YPOT) — Print out composition of direct potential.

**Routine YZK**
`(IPD,K,I,J)`

This routine evaluates Hartree $Y$- and $Z$-functions:

$$Y^{(k)}(i,j;r) = Z^{(k)}(i,j;r) + W^{(k)}(i,j;r) \tag{4.22}$$

where

$$Z^{(k)}(i,j;r) = \int_0^r \left(\frac{s}{r}\right) \left(P_i(s)P_j(s) + Q_i(s)Q_j(s)\right) ds \tag{4.23}$$

and

$$W^{(k)}(i,j;r) = \int_r^\infty \left(\frac{r}{s}\right)^{k+1} \left(P_i(s)P_j(s) + Q_i(s)Q_j(s)\right) ds \tag{4.24}$$

The $Y$-function is tabulated in common block /TATB/ in array `TB`, the $Z$-function in array `TA`.

The correction to $Z^{(0)}$ is in the manner of Froese Fischer (1977 [6]) p.235.

Routine(s) called: DRAW

**Routine ZEFE**
`(IPD,J,ZE)`

This routine calculates an effective charge `ZE` such that in a Coulomb field of charge `ZE` the eigenvalue of orbital `J` is equal to the input value `E(J)`.

Routine(s) called: none

**Routine ZEFR**
`(IPD,J,R,ZE)`

This routine calculates an effective charge `ZE` such that in a Coulomb field of charge `ZE` the mean radius $< r >$ of orbital `J` is equal to the input value `R`.

Routine(s) called: RBAR

**Routine ZKF**
`(K,I,J)`

This routine evaluates Hartree $Z$-functionals:

$$Z^{(k)}[f(r);r] = \int_0^r \left(\frac{s}{r}\right)^k f(s)\,ds \tag{4.25}$$

The Z-functional is tabulated in common block /TATB/ in array `TB`. The $f$-function is assumed tabulated in array `TA`.

# Chapter 5

# GRASP$^2$ — NJGRAF routines

These routines are the NJGRAF package as supplied with GRASP[2].

**Routine NJGRAF**
`(IPD,IBUG3,RECUP,IGEN,FAIL)`

Program to calculate a general recoupling coefficient.

These routines have been take from a CPC library program (cat. no. ABBY). We refer the user to Bar-Shalom and Klapisch (1988 [3]) for details. Incorporation of NJGRAF into GRASP has necessitated certain changes. These are described in detail in the routine headers.

This version is slightly modified. The changes are as follows:

1. Parameter `IGEN` has been included in the argument list: `IGEN`= 0 normal call to NJGRAF
   `IGEN`= −1 GENSUM is not called
2. The contents of common blocks /ARGU/ and /SUMARG/ are used by GENSUM to calculate the recoupling coefficient. These common blocks have been removed from GENSUM. Their contents are passed to GENSUM through the argument list instead, so that NJGRAF can be called to set up formulae for both the direct and exchange cases in COR and BREIT.
3. Extra dimension tests have been included in routines NJGRAF, PRINTJ, SPRATE, VAR and ZEROJ. These are discussed below.
4. An extra routine RDIAG has been introduced to remove an extended DO loop from GENSUM, to conform with the FORTRAN 77 standard.

- `IPD`
  output datastream number
- `IBUG3`
  debug prints in NJGRAF and GENSUM if 1

Description of some common blocks.

- COMMON BLOCK COUPLE
  - `M`
    the total number of angular momentum values in the initial and final states
  - `N`
    the number of basic angular momentum values that are coupled
  - `J1(I)`, `I=1,M`
    the angular momentum values stored as $2J + 1$
  - `J2(I,J)`, `I=1,(N-1)`, `J=1,3`
    the position in the `J1` array of the initial state triads

- `J3(I,J)`, `I=1,(N-1)`, `J=1,3`
  the position in the `J1` array of the final state triads
- `FREE(I)`, `I=1,M`
  if `FREE(I)` = .TRUE., no reference is made to the value of `J1(I)` when establishing a formula in NJGRAF. GENSUM may then be called for repeated occurrences of this formula with differing values of `J1(I)`. If `J1(I)` does not vary between calls to GENSUM then `FREE(I)` should be set .FALSE. so that zero branches can be removed before the formula is established.

- COMMON BLOCK ARGU
  - J6C
    the number of elements in the K6 array
  - J7C
    the number of elements in the K7 array
  - J8C
    the number of elements in the K8 array
  - J9C
    the number of elements in the K9 array
  - JWC
    the number of columns in the KW array
  - J6(I), I = 1,J6C
    each entry corresponds to a factor SQRT(2J+1) in RECUP. The value of J6 gives position in J1 array where J value is found
  - J7(I), I = 1,J7C
    each entry corresponds to a factor (-1)**J in RECUP
  - J8(I), I = 1,J8C
    each entry corresponds to a factor (-1)**(2J) in RECUP
  - J9(I), I = 1,J9C
    each entry corresponds to a factor (2J+1)**(-0.5) in RECUP
  - KW(I,J), I = 1,6, J = 1,JWC
    each column corresponds to a Racah coefficient in RECUP
  - JDEL
    the number of delta functions
  - LDEL(I,J), J = 1,2
    the arguments of the delta functions
  - SUMVAR(I)
    .TRUE. for ang. mom. I (a summation variable)
  - MP
    the index of the last variable

The arrays J6, J7, J8, J9 and KW, are evaluated by NJGRAF. The summation over the variables in J6, J7, J8, J9 and KW, and the evaluation of RECUP is carried out in GENSUM. GENSUM can be re-entered directly to evaluate different recoupling coefficients with the same structure by just altering the numbers in the J1 array.

This is the main program. It handles all the analysis of the recoupling coefficient without referring explicitly to the values of angular momenta which are in J1(J), except for zero in case FREE = .FALSE. . Like NJSYM it prepares arrays of arguments for phase factors, (2*J+1) factors and 6$J$-coefficients to be computed in GENSUM, which can also be called separately when only the numerical values of angular momenta change. These variable angular momenta should be declared FREE(J) = .TRUE., so that the formula prepared for GENSUM should be correct when J1 is not zero. FAIL will be TRUE when the recoupling coefficient is zero because of unsatisfied delta or other similar causes.

This version holds the array dimensions in PARAMETER statements. The dimensions are labelled:

- MANGM
  dimension of the J1 and FREE arrays in /COUPLE/, and the first dimension of the LINE and LCOL arrays in /TREE/. Also the dimension of the SUMVAR array in /ARGU/, and of the INVER array in routine SPRATE. It is tested for M on entry to NJGRAF, and for MP in routine SPRATE.

- MTRIAD
  dimension of the J2 and J3 arrays in /COUPLE/. The dimensions of these arrays are checked on entry to NJGRAF in addition MTRIAD sets the dimension of the JSUM6 array and the first dimension of the JSUM4 and JSUM5 arrays in /SUMARG/. Also gives the dimensions of some temporary working arrays in SPRATE and GENSUM. In these cases MTRIAD sets the maximum number of summation variables in any particular sum, which is tested in SPRATE.

- M2TRD
  (=2∗MTRIAD) dimension of the J23, ARROW and TABS arrays in /TREE/. Also the dimension of the NPOINT array in /GRAPH/.

- M4TRD
  (=4∗MTRIAD) dimension of the JDIAG, ARR, IL and IH arrays in /GRAPH/, and of the IAL array in /BUILD/.

- M3MNGM
  dimension of the J6 array in /ARGU/, tested in SPRATE dimension of the J7 array in /ARGU/, tested in SPRATE dimension of the J8 array in /ARGU/, tested in SPRATE

- MANGMP
  dimension of the J9 array in /ARGU/, tested in SPRATE MANGMP also sets the dimension of the J6P, J7P, J8P and J9P arrays in /SUMARG/, and of the JNS array in routine VAR. The dimension of the JNS array is tested in VAR.

- M6J
  dimension of the JW(OR KW) and LDEL arrays in /ARGU/, and of the JWORD and INV6J arrays in /SUMARG/. Also the second dimension of the JSUM4 and JSUM5 arrays in /SUMARG/. In addition it gives the dimensions of a number of temporary working arrays in routines SPRATE and GENSUM. M6J is tested in SPRATE.

- MFACT
  dimension of the factorial array GAM in /FACTS/.

- MSUM
  dimension of the NBJ, NB6J, K6CP, K7CP, K8CP and K9CP arrays in /SUMARG/. MSUM is the maximum number of sums allowed, and is tested in routine SPRATE.

- MTAB
  dimension of the JTAB array in routine PRINTJ. MTAB is tested in PRINTJ.

- MZERO
  dimension of the JZERO array in /ZER/. MZERO is tested in routine ZEROJ.

**Routine BUBBLE**
`(IPD,IBUG3,JPOL,FAIL)`

This routine reduces a circuit of order 2, giving delta function and phase factors.

**Routine CHANGE**
`(L,K)`

This routine exchanges the free ends in either first or last triad of `JDIAG`.

**Routine CHVAR**
`(JP,NBC,KBC,JT,JINV,NSUM)`

This routine changes the order of summation variable to be able to perform separately the summations in GENSUM.

**Routine CUT1L**
`(IPD,IBUG3,FAIL)`

Cut on one line, that was left as a free end in `JDIAG`. Puts corresponding delta in `J23`.

**Routine CUT2L**
`(IPD,IBUG3,FAIL)`

Cut on two lines that were left as free ends in `JDIAG`. Puts corresponding delta in `J23`.

**Routine CUTNL**
`(IPD,IBUG3,FAIL)`

This routine examines the case where there are more than two free ends, but they are contiguous, so that the graph can be cut without destroying the flat structure.

**Routine DELTA**
`(IPD,IBUG3,JA,JB,FAIL)`

Test for delta(JA,JB). If they are summation variables, the second is changed into the first everywhere. If they are fixed, their value is checked, and FAIL put to .TRUE. if they differ.

**Routine DIAGRM**
(IPD,IBUG3,JUMP)

This routine builds up a flat diagram from the triads J23 and places them in JDIAG. Arrows are in ARR (integer). The diagram is built so as to maximise the number of triads involved, within a one-step-forward-check process. If the diagram does not include all the NBTR triads, it will have 'free ends'. JDIAG has dimension double that of J23, because the path may proceed either way.

**Routine GENSUM**
(IPD,IBUG3,
J6C,J7C,J8C,J9C,JWC,J6,J7,J8,J9,JW,JDEL,LDEL,
MP,J6P,J7P,J8P,J9P,JWORD,NLSUM,NBJ,NB6J,
K6CP,K7CP,K8CP,K9CP,JSUM4,JSUM5,JSUM6,INV6J,
RECUP)

This routine carries out the summation over coefficients defined by the arrays J6, J7, J8, LDEL and JW to give RECUP. The entry is either made from NJGRAF or directly assuming that the arrays J6, $\cdots$, JW have already been determined by a previous entry to NJGRAF and that the summation is required for another set of $J$ values defined by the array J1. RECUP is the recoupling coefficient.

Routine(s) called: DRACAH, RDIAG.

**Routine INTAB**

This routine called at the end of DIAGRM, fixes the arrays IH and IL - so to speak hardware and logical addresses of triads in JDIAG. Also determines the number of free ends NFREE and their location ITFREE.

**Routine LOLPOP**
(IPD,IBUG3,FAIL)

This routine reduces a loop with one line and one node in the flat graph.

Routine(s) called: DELTA, PHASE2

**Routine NEIBOR**
(LC,L1,L2)

This routine gives the positions of the other two arguments in the triad.

**Routine ORDTRI**
(IPD,IBUG3)

This routine orders the triads which were left with free ends as consequence of cutting, so that the new graph will start there.

**Routine OTHERJ**
(LIN,J,LO,LCO,K)

Gives the other triad where a given J occurs and its position.

**Routine PHASE**
(L,JM,NDIM)

This routine evaluates the phase factor arising from non-cyclic permutation of arguments in triad L. JM may be either J23 or JDIAG.

**Routine PHASE2**
(J)

This routine adds a phase factor (-1)**2J.

**Routine POLYGN**
(IPD,IBUG3,JPOL)

This routine reduces a circuit of arbitrary order NC. It exchanges nodes on the flat diagram until the distance on the axis between nodes equals one. Each exchange introduces a summation variable and a $6j$-symbol. The circuit has a maximum of NPART=2 disconnected parts on the axis.

**Routine PRINTJ**
(IPD,IBUG3,NAMES,JP)

This routine prints intermediate results in standard form from wherever it is called.

**Routine RDIAG**
(I,J,IK1,IK2,ICHAN,MAT,JSUM,J12)

This routine is called by GENSUM to establish the range of values of the summation variables. This replaces an extended range DO loop in GENSUM, to conform with the FORTRAN 77 standard.

Routine(s) called: none

**Routine SEARCH**
(IPD,IBUG3,FIND)

This routine locates circuits or loops of order NC. NPOINT(NC) are the indices of the points(triads) pertaining to the first such loop found. NPART is the number of separate parts (groups of contiguous points) on the axis of the flat graph. IPARTS is the number of points in the smallest part. IPARTL is the number of points in the largest part. The routine finds all the possible loops of order 3 and 4. For NC$\geq$ 5, it looks for only those who are partitioned in NPART$\leq$ 2 which can eventually reduce to a loop of order 4 without breaking the basic structure of the flat graph. ICROSS= $-1$, if lines cross.

**Routine SETDM**

This routine sets dimensions of arrays.

**Routine SETTAB**
(IPD,IBUG3,FAIL)

This routine builds up the unstructured graph. Sets the array J23, containing the two lists of original triads J2 and J3, and the corresponding arrows on the angular momenta

lines. Also establishes the numerical and phase factors connecting recoupling coefficient and graphs, according to Yutsis, Levinson and Vanagas. For this purpose determines the total $J$.

**Routine SPRATE**
`(IPD,M)`

This routine prepares the information to be transferred to GENSUM for numerical evaluation.

**Routine SQUARE**

This routine reduces a circuit of order 4 in the two cases which are left over by POLYGN, namely two disconnected groups of two points and one group of two points plus the two ends of the axis. In the latter, the end of the axis is transferred to the beginning. In this process, one summation variable and two $6j$-symbols are introduced.

**Routine TRDEL**
`(JJ1,JJ2,JJ3,NBN,FAIL)`

This routine tests for triangular delta. If not satisfied `FAIL`=.TRUE. .

**Routine TRIANG**
`(FAIL)`

This routine reduces a triangle having one apex at either end of the axis of the flat diagram. This introduces one $6j$-symbol and some phase factors.

**Routine VAR**
`(IPD,JN,JNS,JNC,JNSC,JBC,SUMVAR,MP,M,INVER)`

This routine tests for variable character and put in `JNS` if yes, and `JN` now contains 0.

**Routine WAY**
`(L,KA,KB,ICH,NB)`

Tests one step forward if the way is free. First and second arguments are interchanged or not according to `ICH`$=-1$ or $+1$.

**Routine ZEROJ**
`(IPD,IBUG3,J,JZ,FAIL)`

This routine suppresses one line and two nodes of the unstructured graph and introduces zeros in the triads `J23`. As a consequence the other two arguments of the triad are put equal. If there was already a zero in the triad which is changed, it is a special case.

**Routine DRACAH**
`(I,J,K,L,M,N,RAC)`

This routine calculates Racah coefficients. The arguments `I`, `J`, `K`, `L`, `M`, `N` should be twice their actual value. It works for integer and half-integer values of angular momenta. The routine makes use of the `GAM` array, thus routine FACTT must be called before this routine is used. Written by NS Scott.

**Routine FACTT**

This routine calculates the logs of factorials required by the Racah coefficient routine DRACAH. Written by NS Scott.

# Chapter 6

# GRASP$^2$ — system routines

These are system dependent routines used in GRASP[2].

**Routine CALEN**

CALEN returns in common block TITL the time and date at execution of program.

**Routine TIMER**
`(I)`

This routine calculates either the CPU time which is left or has been used. It has three modes:

- `I`=−1 used to initialise the time stored in TIME1
- `I`=0 writes out time used in seconds since last call
- `I`=1 returns in TIME2 the time left in seconds

# Chapter 7

# References

# Bibliography


[1] M Abramowitz and IA Stegun *Handbook of Mathematical Functions* (Dover, 1965 )

[2] AI Akhiezer and VB Berestetskii *Quantum Electrodynamics* (US Atomic Energy Commission, 1957 — translation of 1953 Russian original)

[3] A Bar-Shalom and M Klapisch *Comput. Phys. Commun.* **50** (1988) 375–393
*NJGRAF: an efficient program for calculation of general recoupling coefficients by graphical analysis, compatible with NJSYM*

[4] DM Brink and GR Satchler *Angular Momentum* (3rd edition, Oxford University Press, 1993 )

[5] C Froese *Can. J. Phys.* **41** (1963) 1895

[6] C Froese Fischer *The Hartree-Fock Method for Atoms* (Wiley, 1977 )

[7] C Froese Fischer *Comp. Phys. Rep.* **3** (1986) 273–326

[8] LW Fullerton and GA Rinker Jr *Phys. Rev. A* **13** (1976) 1283–1287

[9] IP Grant *Adv. Phys.* **19** (1970) 747–811
*Relativistic calculation of atomic structures*

[10] IP Grant *Comput. Phys. Commun.* **4** (1972) 377–381
*CFPJJ – fractional parentage coefficients for equivalent electrons in $jj$-coupling*

[11] IP Grant *Comput. Phys. Commun.* **5** (1973) 263–282
*A general program to calculate angular momentum coefficients in relativistic atomic structure*

[12] IP Grant *J. Phys. B: At. Mol. Phys.* **7** (1974) 1458–1475
*Gauge invariance and relativistic radiative transitions*

[13] IP Grant *Comput. Phys. Commun.* **11** (1976) 397–405
*A program to calculate angular momentum coefficients in relativistic atomic structure — revised version*

[14] IP Grant *Comput. Phys. Commun.* **14** (1978) 312

[15] IP Grant, BJ McKenzie, PH Norrington, DF Mayers and NC Pyper *Comput. Phys. Commun.* **21** (1980) 207
*An atomic multi-configurational Dirac-Fock package*

[16] IP Grant and BJ McKenzie *J. Phys. B: At. Mol. Phys.* **13** (1980) 2671–2681

[17] FB Hildebrand *Introduction to Numerical Analysis* (McGraw-Hill, 2nd edition, 1974 )

[18] WR Johnson and KT Cheng *J. Phys. B: At. Mol. Phys.* **12** (1979) 863

[19] S Klarsfeld and A Maquet *Physics Letters* **43B** (1973) 201

[20] AH MacDonald and SH Vosko *J. Phys. C: Solid State Phys.* **12** (1979) 2977–2990

[21] BJ McKenzie, IP Grant and PH Norrington *Comput. Phys. Commun.* **21** (1980) 233
*erratum* **23** (1981) 222
*A program to calculate transverse Breit and QED corrections to energy levels in a multi-configurational Dirac-Fock environment*

[22] PJ Mohr *Atomic Data and Nuclear Data Tables* **29** (1983) 453

[23] WH Press, BP Flannery, SA Teukolsky and WT Vetterling *Numerical Recipes* (Cambridge University Press, 1986 )

[24] NC Pyper, IP Grant and N Beatham *Comput. Phys. Commun.* **15** (1978) 387–400
*A new program for calculating matrix elements of one-particle operators in $jj$-coupling*

[25] MV Ramana and AK Rajagopal *Phys. Rev. A* **24** (1981) 1689–1695

[26] H Rutishauser *Numer. Math.* **9** (1966) 1–10
*The Jacobi method for real symmetric matrices*

[27] JE Sienkiewicz and WE Baylis *J. Phys. B: At. Mol. Phys.* **20** (1987) 5145–5156