ELSEVIER

Computer Physics Communications 92 (1995) 290–420

Computer Physics Communications

RMATRX1: Belfast atomic *R*-matrix codes

Keith A. Berrington[a], Werner B. Eissner[b], Patrick H. Norrington[a]

[a] *The School of Mathematics and Physics, The Queen's University of Belfast, Belfast BT7 1NN, Northern Ireland*

[b] *Fakultaet für Physik & Astronomie, Ruhr-Universität, D-44 780 Bochum, Germany*

Received 25 November 1994

Abstract

RMATRX1 is a general program to calculate atomic continuum processes using the *R*-matrix method, including electron-atom and electron-ion scattering, and radiative processes such as bound-bound transitions, photoionization and polarizabilities. Calculations can be either in *LS*-coupling, or in an intermediate-coupling scheme by including terms of the Breit-Pauli Hamiltonian. This version supersedes all previous versions, is well-optimised, and contains many new and improved features.

Keywords: Atomic; Electron atom scattering; Electron ion scattering; Photoionization; Polarizability; *R*-matrix

NEW VERSION SUMMARY

Title of program: RMATRX1

Catalogue number: ADCP

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Reference to previous versions: Comput. Phys. Commun. 8 (1974) 149–198; 14 (1978) 367–412; 25 (1982) 347–387 (N.B.: the new version has more features than these).

Does the new version supersede the original program?: Yes

Licensing provisions: none

Computer for which the new version is designed and others on which it has been tested:
1. Cray Y-MP EL, Queen's University Belfast
2. Cray Y-MP I/8128, Rutherford Appleton Laboratory
3. HP apollo 700, Queen's University Belfast
4. VAX 9000, Queen's University Belfast
5. Dell 450L, Queen's University Belfast

Operating systems under which the new version has been tested: UNIX (1 to 3), VMS (4), extended DOS (5)

Programming language used in the new version: FORTRAN 77

Memory required to execute with typical data: 2 MWords minimum. Since the dimensions can be preprocessed it is possible to run small calculations in less memory.

No. of bits in a word: 64

No. of processors used: 1

Has the code been vectorised?: Yes. Time-critical loops have been written in vectorisable form: this is particularly true in program modules STG1 and STGH. In addition, routine MDIAG (matrix diagonalization) in STGH contains parallelisation (Cray micro-tasking) options – see listing for further details.

Peripherals used: scratch disk store; permanent disk store

No. of lines in distributed program, including test data, etc.: 49896

CPC Program Library subprograms used: none (they are incorporated into the present package)

Keywords: atomic, electron atom scattering, electron ion scattering, photoionization, polarizability, R-matrix

Nature of physical problem
This program uses the R-matrix method to calculate electron-atom and electron-ion collision processes, with options to calculate radiative data, photoionization etc. Calculations can be either in LS-coupling or in an intermediate-coupling scheme.

The program is based on two earlier CPC programs [1,7], with extensions by the Opacity Project [2,8] and the Iron Project team [6].

Method of solution
The R-matrix method is used [3,4,5]. The code is written for an LS-coupling scheme, with options to include Breit-Pauli terms in the Hamiltonian and recouple the matrices to an intermediate-coupling scheme.

Reasons for new version, and summary of revisions
- incorporate new sections for calculating radiative data
- merge the LS-coupling and Breit-Pauli versions
- implement new algorithms, better numerical procedures, etc.
- improve optimisation (e.g. vectorisation, reducing I/O)
- upgrade to FORTRAN 77
- allow preprocessing of arrays

Restrictions on the complexity of the problem
The main purpose of the present publication is to publish the internal region modules. The external region module (STG4) is rather inefficient and should be replaced in production runs.

Typical running time
The test runs take about 2 minutes on a Cray Y-MP EL. However this is a small atypical calculation.

The running time in a realistic calculation tends to be dominated by three processes:
(a) calculating angular integrals in STG2, particularly for open d-shell targets;
(b) diagonalizing large Hamiltonian matrices in STGH, an n^3 process, where n is the size of the Hamiltonian and is proportional to the number of channels and basis terms.
(c) solving the coupled equations in the external region, depends on the number of scattering energies required as well as the number of channels.

Unusual features of the program
Dimensions can be reset by preprocessing.

References
[1] K.A. Berrington, P.G. Burke, M. Le Dourneuf, W.D. Robb, K.T. Taylor and Vo Ky Lan, Comput. Phys. Commun. 14 (1978) 367-412.
[2] K.A. Berrington, P.G. Burke, K. Butler, M.J. Seaton, P.J. Storey, K.T. Taylor and Yu Yan, J. Phys. B: At. Mol. Phys. 20 (1987) 6379-97.
[3] P.G. Burke and K.A. Berrington, Atomic and Molecular Processes, an R-matrix Approach (Institute of Physics Publ. Bristol, 1993).
[4] P.G. Burke and W.D. Robb, Adv. At. Mol. Phys. 11 (1975) 143-214.
[5] P.G. Burke, A. Hibbert and W.D. Robb, J. Phys. B: At. Mol. Phys. 4 (1971) 153-61.
[6] D.G. Hummer, K.A. Berrington, W. Eissner, A.K. Pradhan, H.E. Saraph and J.A. Tully, Astron. Astrophys. 279 (1993) 298-309.
[7] N.S. Scott and K.T. Taylor, *Comput. Phys. Commun.* 25 (1982) 347-87.
[8] M.J. Seaton, J. Phys. B: At. Mol. Phys. 20 (1987) 6363-78.

LONG WRITE-UP

Contents

1. Introduction

1.1. Background

We are concerned here with modelling the interaction of electrons and photons with isolated atoms and their ions.

There is a continuing demand for a large variety of high quality atomic data to understand and predict laboratory or astrophysical processes. This is due in part to the ever greater precision of modern experimental and observational facilities; in part to the desire for greater understanding of plasma properties, and of the results of sophisticated experiments. It appears that very little of the relevant data can be provided in controlled laboratory conditions. This gives a tremendous motivation to exploit powerful computational methods, such as the R-matrix method, and supercomputers to provide such data. There is also the satisfaction of being able to solve in an *ab initio* way the quantum mechanical equations underlying natural phenomena, in order to understand these very basic interactions.

As will be seen, an important advantage of the R-matrix method is that it provides an excellent basis for describing many types of atomic process.

The atomic R-matrix programs described here have a long history. In order to give some idea of the evolution and application of these programs, let us outline some of the publication milestones.

1974 First publication of R-matrix package (LS-coupling version) by Berrington et al. [12], based on theory developed by Burke and Seaton [23], Burke, Hibbert and Robb [27], Burke and Robb [22]. These three references are essential reading for the beginner; the latter reference being particularly readable.

1978 Modified R-matrix package published by Berrington et al. [13].

1982 Publication of Breit-Pauli version of R-matrix programs by Scott and Taylor [73], based on theory developed by Scott and Burke [71].

1987 First publications from the international Opacity Project by Seaton [78] and Berrington et al. [11], which describe extensions to the 1978 version for calculating radiative data for stellar opacities. The Project was reviewed in a book by the Opacity Project team [64], containing an introduction, 21 key papers, and tables of energy levels and oscillator strengths.

1993 First publication from the international Iron Project by Hummer et al. [54], describing extensive modifications for calculating excitation data for the iron group of atoms and ions.

1993 The publication of a comprehensive review of the R-matrix method and its applications in a book edited by Burke and Berrington [21], which contains introductory material, a bibliography of 547 R-matrix publications, with 27 of the key papers reprinted.

It is clear that some time has elapsed since the programs themselves were last published. Many changes have been made since then, the most major of which have been
- the writing of new sections for calculating radiative data,
- the merging of the LS-coupling and Breit-Pauli versions,
- the implementation of new algorithms.

There is thus a consistent and general atomic R-matrix computer package. Countless minor modifications have been made: to improve the numerical procedures, minimise use of intermediate disk files, improve optimisation on vector supercomputers, upgrade to FORTRAN 77, allow preprocessing of array dimensions, etc. . A degree of stability has now been achieved, moreover these programs are in use by workers world-wide; it is therefore timely to produce a new write-up and publish the current package, known as RMATRX1.

It should be emphasised that program development has been continuous over time, with many people involved. Although the three authors of this paper have been involved in these developments, and have put together this write-up, we wish to acknowledge the contribution of others. In particular we mention Professors Phil Burke and Mike Seaton, who have played a major role in developing the R-matrix method.

The remainder of this section summarises the atomic processes which can be treated in the present computer codes, and introduces some notation and the choice of units. Section 1.2 discusses the non-relativistic formulation of R-matrix theory, with particular emphasis on electron collisions; Section 1.3 discusses radiative processes; Section 1.4 discusses extensions to include relativistic effects; Section 1.5 discusses using a model potential; and Section 1.6 describes program layout and some programming details.

Each module in RMATRX1 (STG1, STG2, RECUPD, STGH, STG4 and STGLIB) is then described in detail in Sections 2–7, while Section 8 describes the preprocessing of dimensions; Section 9 presents a glossary; Section 10 summarises additional programs which interface with the present codes; Section 11 describes the test runs.

1.1.1. Processes

This document describes the computational implementation of the atomic R-matrix method for the following processes (A refers to an atom or ion, A_i its initial state, A_f its final state):
- elastic and inelastic electron scattering from atoms and ions

$$\mathrm{e}^- + A_i \rightarrow \mathrm{e}^- + A_f \tag{1}$$

- photoionization of atoms and ions

$$h\nu + A \rightarrow \mathrm{e}^- + A^+ \tag{2}$$

- properties of the $(\mathrm{e}^- + A)$ system

  e.g. resonances, bound state energies, oscillator strengths and polarizabilities.

Let N be the number of electrons in A_i and A_f in Eq. (1). This is our atomic or ionic *target*. Less obviously, in a photoionization calculation, we use the word 'target' when referring to the *final* state (A^+ in Eq. (2)), or the N-electron core in a bound-state or polarizability calculation. The initial state in Eq. (2) is of course a bound state of the $(N+1)$-electron system.

At low energies, the probability of a collision process can be dominated by transitions involving the temporary capture of the colliding electron, followed by autoionization, e.g.

$$e^- + A_i \rightarrow (A^-)^* \rightarrow e^- + A_f$$

where the asterix indicates $(N+1)$-electron *resonance* states. It is the description of such processes that arises so naturally from using R-matrix techniques.

1.1.2. Notation and choice of units

Let us first review some notation, and define our units.

A *configuration* of an atom or ion specifies a particular distribution of the electrons among atomic orbitals. Light atoms and ions are well described by Russell-Saunders (LS) coupling, where the orbital (L) and spin (S) angular momenta are assumed to be separately conserved, as is the parity (π). Thus an *LS-term* is a specific $SL\pi$ target state. Introducing the spin-orbit interaction, necessary for an accurate description of heavier atoms and ions, splits the LS-term energies into fine-structure *levels* of symmetry $J\pi$, where J is the total angular momentum.

The equations are expressed in atomic units (a.u.) as introduced by Hartree [49].

The unit of length is the Bohr radius $a_0 = 5.29177 \times 10^{-11}\,\text{m}$

The unit of energy is the Hartree energy $E_\text{h} = 4.35975 \times 10^{-18}\,\text{J}$

Energies are often expressed in Rydbergs (Ry) or electron-volts (eV) where

$$E_\text{h} = 2\,\text{Ry} \approx 27.2114\,\text{eV}$$

Using Ry the energy levels of the hydrogen atom are $E_n = -1/n^2$ and the energy of a free electron is k^2 where k is the wavenumber in a.u. . Boltzmann's constant, k_B, is such that, for a temperature T in K,

$$k_B T \approx (T/157890)\ \text{Ry}$$

In a.u. the speed of light c takes the value of 137.036 (the reciprocal of Sommerfeld's 'fine-structure constant' α).

Cross sections have units of area, and are sometimes quoted in units of $\pi a_0^2 \approx 88 \times 10^{-22}\,\text{m}^2$. Photoionization cross sections are traditionally quoted in Megabarns ($1\text{Mb} = 10^{-22}\,\text{m}^2$).

The values of physical constants are taken from CODATA [35].

1.2. Non-relativistic R-matrix theory

In this section we review R-matrix theory with particular emphasis on non-relativistic electron scattering from atoms and ions.

1.2.1. Equation of motion

The scattering system consists of an atomic target with N electrons and an additional colliding electron. It is assumed that the behaviour of this system is determined solely by the electromagnetic interaction between the charged particles. In quantum mechanics, all information on the system is contained in the wavefunction.

We therefore seek solutions Ψ, with appropriate boundary conditions, for the time independent Schrödinger equation

$$H^{N+1}\Psi = E\Psi \tag{3}$$

where E is the total energy.

For light atoms and ions, where relativistic effects can be neglected, the $(N+1)$-electron Hamiltonian (in a.u.) is given by

$$H^{N+1} = \sum_{n=1}^{N+1} \left(-\tfrac{1}{2}\nabla_n^2 - \frac{Z}{r_n} + \sum_{m>n}^{N+1} \frac{1}{r_{nm}} \right) \tag{4}$$

where $\boldsymbol{r}_n$ is an electronic radius vector drawn from the atomic nucleus with atomic number Z and $r_{nm} = |\boldsymbol{r}_n - \boldsymbol{r}_m|$ an inter-electronic distance. It is assumed that the nucleus is infinitely heavy and is a point charge.

The first two terms on the right-hand side of Eq. (4) are a sum over the electron kinetic energy and electron-nucleus Coulomb attraction: the so-called *one-electron* terms. The final term on the right is a summation over the Coulomb repulsion between pairs of electrons: the so-called *two-electron* term.

In the calculation of matrix elements, we will be expanding $1/r_{nm}$ in terms of spherical harmonics:

$$\frac{1}{r_{nm}} = \sum_{\lambda\mu} \left(\tfrac{4\pi}{2\lambda+1}\right) Y_\lambda^{\mu *}(\hat{r}_n) Y_\lambda^{\mu}(\hat{r}_m) \frac{r_<^\lambda}{r_>^{\lambda+1}} \tag{5}$$

where $r_<$ & $r_>$ = min & max (r_n, r_m), and the angular integrals are evaluated using standard techniques including the Wigner-Eckart theorem.

Solutions Ψ of Eq. (3) are constructed as products of one-electron functions in spherical polar coordinates — which makes configuration mixing a requirement. Later we shall see that such coordinates reduce Eq. (3) to Hartree-Fock type radial equations.

1.2.2. Target states

Before considering the full $(N+1)$-electron system, let us first examine techniques for establishing a wavefunction for the N-electron target.

We introduce a set of target states, and possibly pseudo-states, Φ_i and their corresponding energies E_i^N by the equation

$$\langle \Phi_i \mid H^N \mid \Phi_j \rangle = E_i^N \delta_{ij} \tag{6}$$

where we use the bra and ket notation, implying an integration over all electronic coordinates over all space. Here H^N is the target Hamiltonian defined by Eq. (4) with $N+1$ replaced by N. These are antisymmetric $LS\pi$ states.

These states are usually written as a configuration-interaction (CI) expansion in terms of some basis configurations ϕ_i by

$$\Phi_i(x_1 \dots x_N) = \sum_k b_{ik}\, \phi_k(x_1 \dots x_N) \tag{7}$$

where $x_\nu \equiv \boldsymbol{r}_\nu \sigma_\nu = r_\nu\, \hat{r}_\nu, \sigma_\nu$ stands for the spatial position and spin of the νth electron, and the configuration mixing coefficients b_{ik} are determined by diagonalizing the target Hamiltonian in the representation of Eq. (7).

The configurations ϕ_i are constructed from a bound orbital basis usually consisting of self consistent field (SCF) orbitals plus some additional pseudo-orbitals included to model electron correlation effects. For a given ϕ_i these one-electron orbitals $o_{nlm_l}(x_\nu)$ are coupled together to give a function which is completely antisymmetric with respect to the interchange of the space and spin coordinates of any two electrons:

$$\phi_k(x_1 \dots x_N) = (N!)^{-1/2} e_{\nu_1 \dots \nu_N} o_1(x_{\nu_1}) o_2(x_{\nu_2}) \dots o_N(x_{\nu_N})$$

where $N!$ is the number of permutations and e the permutation symbol.

Each one-electron orbital is a product of a radial function, a spherical harmonic, and a spin function:

$$o_{nlm_l}(\boldsymbol{r}, m_s) = \frac{1}{r} P_{nl}(r) Y_l^{m_l}(\theta, \phi) \chi(m_s) \tag{8}$$

The orbitals form an orthonormal set

$$\langle o_{n_i l_i m_{l_i}} \mid o_{n_j l_j m_{l_j}} \rangle = \delta_{n_i n_j} \delta_{l_i l_j} \delta_{m_{l_i} m_{l_j}} \delta_{m_{s_i} m_{s_j}}$$

In particular the radial parts of these orbitals, $P_{nl}(r)$, satisfy the orthonormality relations,

$$\langle P_{n_i l} \mid P_{n_j l} \rangle = \int_0^\infty P_{n_i l}(r) P_{n_j l}(r)\, \mathrm{d}r = \delta_{n_i n_j} \tag{9}$$

The $P_{nl}(r)$ radial functions are input to the R-matrix programs, and are obtained from tabulations such as Clementi and Roetti [34], or from atomic structure packages such as CIV3 (Hibbert [52]) and SUPERSTRUCTURE (Eissner et al. [43]).

The target functions usually die off exponentially at large distances from the atom; the physical interpretation being that the probability of finding any of the N target electrons significantly outside the atom is small, at least at low energies. The localisation of target electrons is crucial to the R-matrix method and is used to define the radius of the sphere $r = a$ delineating the internal region.

1.2.3. Scattering channels

Now let us examine some of the constraints involved in combining the scattering electron with the target.

• Conservation of angular momentum and parity.

The orbital angular momenta of the scattering electron l_i and of the target state may couple vectorially in several different ways to yield the same total L. For example the $2p$ state of hydrogen can couple with an electron of either $l_i = 0$ or $l_i = 2$ to yield a total angular momentum of $L = 1$ (odd parity). These are the so-called scattering electron 'channels', normally indicated by index i.

• Conservation of energy.

E is the total energy and E_i^N the energy of the target state coupled to the ith channel, in a.u. The channel energy of the scattering electron k_i^2, in Ry, is therefore:

$$k_i^2 = 2(E - E_i^N)$$

Note that:

$$\begin{aligned} k_i^2 &> 0 \text{ for open channels} \\ &< 0 \text{ for closed channels} \\ &= 0 \text{ at the threshold energy} \end{aligned}$$

If E_0^{N+1} is the energy of the initial state in a photoionization calculation in a.u., then the electron channel energy k_i^2 is related to the incident photon energy ω in Ry by

$$k_i^2 = \omega + 2(E_0^{N+1} - E_i^N) \tag{10}$$

1.2.4. Partition of configuration space

R-matrix theory starts by partitioning configuration space into two regions by a sphere of radius a centred on the target nucleus (cf. Wigner and Eisenbud [83]).

• In the internal region $r \leq a$, where r is the relative coordinate of the scattered electron and the target nucleus, electron exchange and correlation between the scattered electron and the N-electron target are important and the $(N+1)$-electron collision complex behaves in a similar way to a bound state. Consequently a configuration interaction (CI) expansion of this complex, analogous to that used in bound state calculations is adopted.

- In the external region, $r > a$, electron exchange between the scattered electron and the target can be neglected if the radius a is chosen large enough so that the charge distribution of the target is contained within the sphere. The scattered electron then moves in the long-range multipole potential of the target. This potential is local and the solution in this region can be obtained using an asymptotic expansion or by using perturbation theory.

The two regions are linked by the R-matrix on the boundary.

Implied in the above discussion is that the N-electron target orbitals of Eq. (8) must become vanishingly small in the external region. In practice the boundary radius a is chosen such that

$$| P_{nl}(r = a) | < \delta \tag{11}$$

for all bound orbitals included in the calculation, where δ is some suitably small number.

1.2.5. Internal region

In the internal region the $(N + 1)$th electron must be considered in quantum mechanics as being indistinguishable from the N target electrons. A CI expansion of the wavefunction, as in the bound state problem discussed in Section 1.2.2, is therefore appropriate in this region.

In order to determine the solution of Eq. (3) in the internal region, we introduce energy-independent basis states ψ_k in the internal region, which are expanded in the form

$$\begin{aligned}\psi_k(x_1 \ldots x_{N+1}) = \mathcal{A} \sum_{ij} c_{ijk}\, \overline{\Phi}_i(x_1 \ldots x_N; \hat{r}_{N+1}\sigma_{N+1}) \frac{1}{r_{N+1}} u_{ij}(r_{N+1}) \\ + \sum_j d_{jk}\, \chi_j(x_1 \ldots x_{N+1})\end{aligned} \tag{12}$$

where $\mathcal{A}$ is the antisymmetrisation operator which accounts for electron exchange between the target electrons and the free electron.

The channel functions $\overline{\Phi}_i$ are obtained by coupling the target states Φ_i, defined by Eq. (6) and Eq. (7), with the angular and spin functions of the scattered electron to form states of the total angular momentum and parity.

The quadratically integrable (L^2) functions χ_i, which vanish at the surface of the internal region, are formed from the bound orbitals and are included to ensure completeness of the total wavefunction.

The continuum orbitals u_{ij}, which represent the motion of the scattered electron, are the only terms in Eq. (12) to be non-zero on the surface of the internal region. We will examine these functions in more detail in the next subsection.

The coefficients c_{ijk} and d_{ik} in Eq. (12) are determined by diagonalizing

$$(\psi_k \mid H^{N+1} \mid \psi_{k'}) = E_k \delta_{kk'} \tag{13}$$

where we introduce round brackets in place of the bra and ket notation to indicate the finite range of integration from $r = 0$ to $r = a$. The $(N + 1)$-electron Hamiltonian operator H^{N+1} in this and later equations is projected onto the space of functions ψ_k. To evaluate this, let φ_λ denote collectively the basis functions and the $V_{k\lambda}$ denote collectively the coefficients c_{ijk} and d_{jk} in Eq. (12), which we can rewrite in a more convenient form:

$$\psi_k = \sum_\lambda \varphi_\lambda V_{k\lambda}$$

Then define Hamiltonian matrix elements

$$H_{\lambda\lambda'} = (\varphi_\lambda \mid H^{N+1} \mid \varphi_{\lambda'}) \tag{14}$$

Clearly, the evaluation of these matrix elements proceeds in exactly the same way as in Eq. (6), except that now all radial integrals involving continuum orbitals are taken over the finite range of r. We are therefore able to use the same computer codes. The diagonalization of this matrix yields the eigenvectors $V_{k\lambda}$ (i.e. the c_{ijk} and d_{jk} coefficients in Eq. (12)), together with the eigenvalues E_k in Eq. (13).

We will see later (Section 1.2.7) how we can use this as a basis to expand the total wavefunction in the internal region for any energy.

1.2.6. Continuum orbitals

The choice of the continuum orbitals u_{ij} in Eq. (12) is important. In principle members of any complete set of functions satisfying appropriate boundary conditions at $r = 0$ and $r = a$ can be used. However a careful choice will enable the convergence of the expansion in Eq. (12) to be more rapid. In the work of Burke et al. [27] numerical continuum orbitals satisfying homogeneous boundary conditions were adopted. This approach gives accurate results provided that a correction proposed by Buttle [30], to allow for the truncated expansion, is included.

The continuum orbitals[1] $u_{ij}(r)$ in Eq. (12) for each angular momentum l_i are normally obtained by solving the model single-channel scattering problem

$$\left(\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{l_i(l_i+1)}{r^2} + V_0(r) + k_{ij}^2\right) u_{ij}(r) = \sum_n \Lambda_{ijn} P_{nl_i}(r) \tag{15}$$

subject to the fixed boundary conditions

$$u_{ij}(0) = 0\,, \qquad \left(\frac{a}{u_{ij}(a)}\right)\left(\frac{\mathrm{d}u_{ij}}{\mathrm{d}r}\right)_{r=a} = b \tag{16}$$

- The Lagrange multipliers Λ_{ijn} ensure that the continuum orbitals are orthogonal to bound orbitals $P_{nl_i}(r)$ of the same angular symmetry, though Schmidt orthogonalisation can be used as an alternative.
- $V_0(r)$ is a zero-order potential which near the nucleus behaves like $2Z/r$. It is normally chosen to be the static potential of the target.
- k_{ij}^2 are the eigenvalues in Ry.
- a is the radius of the sphere defining the internal region.
- The constant b is arbitrary, and is normally set to zero.

The orbitals defined by Eq. (15) are orthogonal both to themselves and to the bound orbitals:

$$\int_0^a u_{ij}(r)u_{ij'}(r)\,\mathrm{d}r \equiv (u_{ij} \mid u_{ij'}) = \delta_{jj'}$$

$$(u_{ij} \mid P_{nl_i}) = 0 \tag{17}$$

where again we have introduced round brackets to indicate the finite range of integration. It follows from this and Eq. (9) that the orbitals

$$P_{n_{\min}l_i}\,, \ldots, P_{n_{\max}l_i}\,, u_{i1}\,, u_{i2}\,, \ldots \quad \text{with } n_{\min} = l_i + 1 \tag{18}$$

form a complete set of functions over the range $r = 0$ to $r = a$ for each l_i.

Note that the bound and continuum orbitals defined in this way, as well as the ψ_k basis expansion in Eq. (12), are all independent of the total $(N+1)$-electron energy.

[1] The index i labels the angular momentum. For a given i there is a discrete set of solutions which are labelled by index j.

1.2.7. R-matrix

We are now in a position to establish the total wavefunction Ψ in the inner region, and the R-matrix on the boundary, for any total $(N+1)$-electron energy E. (The following theory is from Burke et al. [27] and Burke and Robb [22].)

We assume that the states ψ_k form a basis for the expansion of the total wavefunction Ψ for any energy E in the region where all electron coordinates $r < a$. We can therefore write

$$\Psi = \sum_k A_{Ek}\psi_k \tag{19}$$

Note that the energy dependence is carried through the A_{Ek} coefficients, the ψ_k basis being energy independent.

In order to determine the A_{Ek} we start from the relation

$$(\psi_k \mid H^{N+1} \mid \Psi) - (\Psi \mid H^{N+1} \mid \psi_k) = (E - E_k)(\psi_k \mid \Psi)$$

which follows from Eq. (3), Eq. (13) and Eq. (19). Only the kinetic energy operator contributes to the left hand side of this equation and so we obtain

$$-\tfrac{1}{2}(N+1)\left[(\psi_k \mid \nabla^2_{N+1} \mid \Psi) - (\Psi \mid \nabla^2_{N+1} \mid \psi_k)\right] = (E - E_k)(\psi_k \mid \Psi)$$

We can simplify this equation by noting that the only non-zero contribution occurs when the operator ∇^2_{N+1} acts on the continuum orbitals. Using Eq. (19) we obtain

$$\begin{aligned}-\tfrac{1}{2}\sum_{ijk'} A_{Ek'}&\left[(\overline{\Phi}_i w_{ik}(r_{N+1}) \mid \nabla^2_{N+1} \mid \overline{\Phi}_j w_{jk'}(r_{N+1})) - (\overline{\Phi}_j w_{jk'}(r_{N+1}) \mid \nabla^2_{N+1} \mid \overline{\Phi}_i w_{ik}(r_{N+1}))\right]\\ &= (E - E_k)(\psi_k \mid \Psi)\end{aligned}$$

where w_{ik} is defined by

$$\tfrac{1}{r}w_{ik}(r) = \tfrac{1}{r}\sum_j c_{ijk}u_{ij}(r) = (\overline{\Phi}_i|\psi_k) \tag{20}$$

Define the reduced radial wavefunction of the scattered electron in channel i at energy E:

$$\tfrac{1}{r}F_i(r) = \tfrac{1}{r}\sum_k A_{Ek}w_{ik}(r) = (\overline{\Phi}_i|\Psi) \tag{21}$$

In Eq. (20) and Eq. (21) the integration is carried out over all electron space and spin coordinates except the radial coordinate r of the scattered electron. Using the orthonormality of the Φ_i, we obtain

$$-\tfrac{1}{2}\sum_i \left[(w_{ik} \mid \frac{\mathrm{d}^2}{\mathrm{d}r^2} \mid F_i) - (F_i \mid \frac{\mathrm{d}^2}{\mathrm{d}r^2} \mid w_{ik})\right] = (E - E_k)A_{Ek}$$

Applying Green's theorem and noting the boundary conditions of Eq. (16), we obtain

$$-\tfrac{1}{2}\sum_i w_{ik}(a)\left(\frac{\mathrm{d}F_i}{\mathrm{d}r} - \frac{b}{a}F_i\right)_{r=a} = (E - E_k)A_{Ek}$$

This allows us to extract the A_{Ek} coefficients:

$$A_{Ek} = \frac{1}{2a}(E_k - E)^{-1}\sum_i w_{ik}(a)\left(a\frac{\mathrm{d}F_i}{\mathrm{d}r} - bF_i\right)_{r=a} \tag{22}$$

Multiplying by w_{jk} and summing over k we obtain, using Eq. (21),

$$F_i(a) = \sum_j R_{ij}(E) \left(a\frac{\mathrm{d}F_j}{\mathrm{d}r} - bF_j \right)_{r=a} \tag{23}$$

where we have introduced the R-matrix , whose elements are defined by

$$R_{ij}(E) = \frac{1}{2a} \sum_k \frac{w_{ik}(a) w_{jk}(a)}{E_k - E} \tag{24}$$

Eq.(23) and Eq. (24) are the basic equations describing the solution of the electron-target scattering problem in the internal region. The surface amplitudes $w_{ik}(a)$ and the poles E_k of the R-matrix are obtained directly from the eigenvectors and eigenvalues of the Hamiltonian matrix defined by Eq. (13). We see that the R-matrix is obtained for all energies by diagonalizing H^{N+1} once for each set of conserved quantum numbers LS and parity π of the electron-target system.

The logarithmic derivative of the reduced radial wavefunction of the scattered electron on the boundary of the internal region is given by Eq. (23), and is to be matched across the boundary to the external region.

1.2.8. Buttle correction

The most important source of error in this method is the truncation of the expansion in Eq. (24) to a finite number of terms. The distant, neglected terms can play an important role in the diagonal elements of the R-matrix where they add coherently. We can include these terms in Eq. (24) by solving the differential equation

$$\left(\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{l_i(l_i+1)}{r^2} + V_0(r) + k_i^2 \right) u_i^0(r) = \sum_k \Lambda_{ijk} P_k(r)$$

This is the same as Eq. (15) but is solved here at the channel energies k_i^2 without applying the boundary condition Eq. (16) at $r = a$. Suppose the R-matrix is calculated from the first $\mathcal{N}$ terms in the continuum expansion. The correction R_{ii}^0 to the diagonal elements of the R-matrix at the energy k_i^2 is then given, using the formula discussed by Buttle [30], by

$$\begin{aligned} R_{ii}^{\mathrm{c}}(\mathcal{N}, k_i^2) &\approx \frac{1}{a} \sum_{j=\mathcal{N}+1}^{\infty} \frac{u_{ij}(a)^2}{k_{ij}^2 - k_i^2} \\ &= \left[\frac{a}{u_i^0(a)} \left(\frac{\mathrm{d}u_i^0}{\mathrm{d}r} \right)_{r=a} - b \right]^{-1} - \frac{1}{a} \sum_{j=1}^{\mathcal{N}} \frac{u_{ij}(a)^2}{k_{ij}^2 - k_i^2} \end{aligned} \tag{25}$$

where $u_{ij}(r)$ and k_{ij} refer to the jth eigensolution of Eq. (15) satisfying the boundary conditions of Eq. (16), and u_i^0 is the solution for channel energy k_i^2 Ry. We therefore use the Buttle corrected R-matrix in place of Eq. (24):

$$R_{ij}(E) = \frac{1}{2a} \sum_k \frac{w_{ik}(a) w_{jk}(a)}{E_k - E} + R_{ii}^{\mathrm{c}}(\mathcal{N}, k_i^2)\delta_{ij} \tag{26}$$

For $k_i^2 < k_{ij}^2$, $R_{ii}^{\mathrm{c}}(\mathcal{N}, k_i^2)$ is a continuous and monotonic function of k_i^2 and in practice it is often required for a large number of energy values. An efficient procedure then is to calculate $R^{\mathrm{c}}(\mathcal{N}, k^2)$ for some small number of energies k^2 and to fit to a convenient functional form. Fitting to a quadratic in k^2 has been used but this fails to give a good fit over an extended range of k^2, particularly for negative energies. Instead we normally use Seaton's [77] fitting procedure for the Buttle correction.

1.2.9. External region

The next step in the calculation is to solve the electron-target scattering problem in the external region. In this region the colliding electron is outside the atom, and can be considered distinct from the N target electrons. The total wavefunction is expanded in the form

$$\Psi(x_1 \ldots x_{N+1}) = \sum_i \overline{\Phi}_i(x_1 \ldots x_N; \hat{r}_{N+1}\sigma_{N+1}) \frac{1}{r_{N+1}} F_i(r_{N+1}) \tag{27}$$

where the $\overline{\Phi}_i$ are the same set of channel functions used in Eq. (12), but now no antisymmetrisation is required. Substituting Eq. (27) into Eq. (3) and projecting onto the channel functions yields a set of coupled differential equations satisfied by the reduced radial wavefunctions $F_i(r)$ of the form

$$\left(\frac{d^2}{dr^2} - \frac{l_i(l_i+1)}{r^2} + \frac{2Z}{r} + k_i^2\right) F_i(r) = 2\sum_{j=1}^{n} V_{ij}(r)F_j(r) \qquad i = 1,n \quad (r \geq a) \tag{28}$$

Here n is the number of channel functions retained in the expansions of Eq. (12) and Eq. (27), l_i and k_i^2 are the channel angular momenta and energies, and the potential matrix V_{ij} is given by

$$\begin{aligned} V_{ij}(r) &= \langle \overline{\Phi}_i | \sum_{m=1}^{N} r_{m,N+1}^{-1} | \overline{\Phi}_j \rangle \\ &= \langle \overline{\Phi}_i | \sum_{m=1}^{N} \sum_{\lambda\mu} \left(\tfrac{4\pi}{2\lambda+1}\right) Y_\lambda^{\mu *}(\hat{r}_m) Y_\lambda^{\mu}(\hat{r}_{N+1}) r_m^\lambda | \overline{\Phi}_j \rangle \frac{1}{r^{\lambda+1}} \end{aligned}$$

where we have expanded $r_{m,N+1}^{-1}$ using Eq. (5) with $r_m < r_{N+1} \equiv r$. Defining the long-range potential coefficients a_{ij}^λ in terms of Legendre polynomials as:

$$a_{ij}^\lambda = \langle \overline{\Phi}_i \mid \sum_{m=1}^{N} r_m^\lambda P_\lambda(\cos\theta_{m,N+1}) \mid \overline{\Phi}_j \rangle \tag{29}$$

and noting that $a_{ij}^0 = N\delta_{ij}$ because of the orthonormality of the $\overline{\Phi}_i$, the differential equations Eq. (28) reduce to

$$\left(\frac{d^2}{dr^2} - \frac{l_i(l_i+1)}{r^2} + \frac{2z}{r} + k_i^2\right) F_i(r) = 2\sum_{\lambda=1}^{\lambda_{\max}} \sum_{j=1}^{n} \frac{a_{ij}^\lambda}{r^{\lambda+1}} F_j(r) \tag{30}$$

where $z = Z - N$ is the residual target charge.

Eq.(30) can be integrated outwards subject to the R-matrix boundary conditions of Eq. (23) at $r = a$ and then fitting to an asymptotic expansion. The external region problem is common to all atomic and molecular close-coupling calculations, and the same computer packages can be used.

The boundary conditions at infinity are (see e.g. Berrington et al. [11])

$$F_{ij}(r) \underset{r\to\infty}{\sim} \begin{cases} k_i^{-1/2}(\sin\theta_i\delta_{ij} + \cos\theta_i K_{ij}) & \text{open channels} \\ \exp(-\phi_i)\delta_{ij} & \text{closed channels} \end{cases} \tag{31}$$

where the second index j on F_{ij} distinguishes the n_a linearly independent solutions of Eq. (30), n_a is the number of open channels, and

$$\begin{aligned}\theta_i &= k_i r - \tfrac{1}{2} l_i \pi - \eta_i \ln 2k_i r + \arg \Gamma(l_i + 1 + \mathrm{i}\eta_i) \\ \eta_i &= -\frac{z}{k_i} \\ \phi_i &= |k_i|\, r - \frac{z}{|k_i|} \ln(2\,|k_i|\, r)\end{aligned} \tag{32}$$

Thus we can, in principle, calculate the wavefunction in the external region. However, the solution is not fully determined until we have found the reactance matrix $\boldsymbol{K}$ in Eq. (31). To do this we need to use our knowledge of the wavefunction at $r = a$.

1.2.10. Matching of solutions – with open channels

We have specified the wavefunction in the internal region (Section 1.2.7) and in the external region (Section 1.2.9). Let us now link these two regions to complete the solution.

The internal region wavefunction Eq. (23) becomes, in matrix formulation, with a dot ($\dot{}$) to denote $\mathrm{d}/\mathrm{d}r$:

$$\boldsymbol{F} = a\boldsymbol{R} \cdot \dot{\boldsymbol{F}} - b\boldsymbol{R} \cdot \boldsymbol{F} \qquad (r \le a) \tag{33}$$

To relate the $n \times n$ dimensional R-matrix with the $n_a \times n_a$ K-matrix defined in Eq. (31), we introduce $n + n_a$ linearly independent solutions $s_{ij}(r)$ and $c_{ij}(r)$ of Eq. (30) satisfying the boundary conditions

$$\left.\begin{matrix} s_{ij}(r) \\ c_{ij}(r) \\ c_{ij}(r) \end{matrix}\right\} \underset{r\to\infty}{\sim} \left\{\begin{matrix} \sin\theta_i\,\delta_{ij} & i = 1,n & j = 1,n_a \\ \cos\theta_i\,\delta_{i,j-n_a} & i = 1,n & j = 1,n_a \\ \exp(-\phi_i)\,\delta_{i,j-n_a} & i = 1,n & j = n_a + 1,n \end{matrix}\right. \tag{34}$$

where θ_i and ϕ_i are given by Eq. (32). A number of computer packages are available for obtaining these solutions, for example Crees [37] [2]. We expand the reduced radial wavefunction $F_{ij}(r)$ as a linear combination of these asymptotic solutions:

$$\boldsymbol{F} = \boldsymbol{s} + \boldsymbol{c}\boldsymbol{K} \qquad (r \ge a) \tag{35}$$

Differentiating:

$$\dot{\boldsymbol{F}} = \dot{\boldsymbol{s}} + \dot{\boldsymbol{c}}\boldsymbol{K}$$

Substituting this into Eq. (33), to match the internal and external region solutions on the boundary and so eliminate $\boldsymbol{F}$ and $\dot{\boldsymbol{F}}$:

$$\boldsymbol{s} + \boldsymbol{c}\boldsymbol{K} = a\boldsymbol{R}(\dot{\boldsymbol{s}} + \dot{\boldsymbol{c}}\boldsymbol{K}) - b\boldsymbol{R}(\boldsymbol{s} + \boldsymbol{c}\boldsymbol{K})$$

Solving for $\boldsymbol{K}$, let:

$$\boldsymbol{A} = -\boldsymbol{s} + a\boldsymbol{R}(\dot{\boldsymbol{s}} - \frac{b}{a}\boldsymbol{s}), \qquad \boldsymbol{B} = +\boldsymbol{c} - a\boldsymbol{R}(\dot{\boldsymbol{c}} - \frac{b}{a}\boldsymbol{c})$$

then

$$\boldsymbol{B}\boldsymbol{K} = \boldsymbol{A} \quad \text{or} \quad \boldsymbol{K} = \boldsymbol{B}^{-1}\boldsymbol{A} \tag{36}$$

which completes the evaluation of the reactance matrix $\boldsymbol{K}$. The K-matrix is real and symmetric, and represents the asymptotic form of the entire wavefunction, containing information from both internal and external regions.

[2] We note in passing that in the case of no long-range coupling ($a_{ij}^{\lambda} = 0$), $s_{ij}(r)$ and $c_{ij}(r)$ are diagonal matrices for all r; the analysis in Sections 1.2.10 and 1.3.1 however remains the same.

1.2.11. Electron collision cross section

Scattering observables may be calculated from the reactance matrix $\boldsymbol{K}$. Define the $n_a \times n_a$ S-matrix and T-matrix by the matrix equations

$$S = (\mathbf{1} - \mathrm{i}\boldsymbol{K})^{-1}(\mathbf{1} + \mathrm{i}\boldsymbol{K}), \qquad \boldsymbol{T} = \boldsymbol{S} - \mathbf{1} \tag{37}$$

We label the S- and T-matrix by the total angular momenta and parity, $LS\pi$.

The partial *collision strength* for a transition from an initial target state denoted by $\alpha_i L_i S_i$ to a final target state denoted by $\alpha_j L_j S_j$, where α_i and α_j represent the additional quantum numbers necessary to completely define the target states, is given by

$$\Omega_{ij}^{LS\pi} = \frac{g}{2} \sum_{l_i l_j} | T_{ij} |^2 \tag{38}$$

where the summation is over the channels coupled to the initial and final states, and[3]

$$g = \begin{cases} (2L+1)(2S+1) & \text{for } LS\text{-coupling} \\ 2J+1 & \text{for intermediate-coupling} \end{cases}$$

The total collision strength is given by

$$\Omega_{ij} = \sum_{LS\pi} \Omega_{ij}^{LS\pi}$$

which is symmetric in i and j.

The total *cross section* for this transition is given by

$$\sigma_{i\to j} = \frac{\pi a_0^2}{k_i^2 g_i} \Omega_{ij}$$

Note that while the R-matrix is determined by a single diagonalization in the internal region, the coupled equations Eq. (30) must be solved for $r \geq a$ to yield the solutions F_{ij} and hence the K-matrix and cross section for each energy of interest.

1.3. Radiative processes

In this section we consider the interaction of a single photon, of frequency ω, with an atomic system. We follow the notation of Burke and Taylor [25], Seaton [78] and Berrington et al. [11].

We are particularly concerned in this section in calculations of radiative data for the bound $(N+1)$-electron system. We therefore consider first the matching of internal/external region solutions when all channels are closed (cf. Section 1.2.10). We then give expressions for the dipole matrix, and show how oscillator strength and photoionization calculations proceed.

1.3.1. Matching of solutions – all channels closed

The internal region wavefunction and the external region equations for closed channels are exactly as described in Sections 1.2.7 and 1.2.9. However, in the bound state problem, the wavefunction satisfies different asymptotic boundary conditions from the free-state solutions discussed in Section 1.2.10, and this leads to a different matching condition.

[3] In the case of intermediate-coupling a target state is labelled by $\alpha_i J_i$ where J_i is the total angular momentum.

When all of the channels are closed we can define n linearly independent solutions of the external-region Eq. (30). These satisfy the boundary conditions (cf. Eq. (34)):

$$c_{ij}(r) \underset{r\to\infty}{\sim} \exp(-\phi_i)\,\delta_{ij} \quad i=1,n \quad j=1,n \tag{39}$$

where ϕ_i is given by Eq. (32). We can expand the required solution in terms of these solutions (cf. Eq. (35)):

$$F_i(r) = \boldsymbol{cx} = \sum_{j=1}^{n} c_{ij}(r)x_j \quad i=1,n \quad (r \geq a) \tag{40}$$

The coefficients x_j can then be determined by substituting this expression for F_i into Eq. (33) which leads to the following n homogeneous equations (cf. Eq. (36)):

$$\boldsymbol{cx} = a\boldsymbol{R\dot{c}x} - b\boldsymbol{Rcx} \tag{41}$$

Solving for $\boldsymbol{x}$, let

$$\boldsymbol{B} = \boldsymbol{c} - a\boldsymbol{R}(\dot{\boldsymbol{c}} - \frac{b}{a}\boldsymbol{c})$$

$$\Rightarrow \boldsymbol{Bx} = 0 = \sum_{j=1}^{n} B_{ij}x_j \quad i=1,n \tag{42}$$

we see that these equations have only nontrivial solutions at the negative energy eigenvalues corresponding to bound states of the electron-atom system. The condition for a solution is

$$\det \boldsymbol{B} = 0$$

An iterative procedure for the energy has to be adopted to achieve this matching which involves the use of a special technique to carry out the calculation in the vicinity of R-matrix poles (Burke and Seaton [24], Seaton [75]).

1.3.2. Dipole matrices

Properties associated with the interaction of a photon with an atomic system can be deduced from the dipole matrix involving the initial and final state wavefunctions. Define the dipole length and velocity operators:

$$\boldsymbol{D}_{\rm L} = \sum_n \boldsymbol{r}_n \quad \text{and} \quad \boldsymbol{D}_{\rm V} = -\sum_n \nabla_n \tag{43}$$

where the summation is over all the electrons.

The reduced dipole matrix $(a \,||\, D \,||\, b)$ between states a and b is defined (using the convention of Fano and Racah [45]) as

$$(L_a \,||\, D_l \,||\, L_b) = \frac{(2L_a+1)^{1/2}}{C(L_b l L_a; M_{L_b}\mu)} \langle L_a M_{L_a} \mid D_l^{\mu} \mid L_b M_{L_b}\rangle \tag{44}$$

μ identifies a component of the dipole vector. When $\mu = 0$, the operator on the right-hand side of this equation is either the dipole length or the dipole velocity operator appearing in Eq. (43).

Bound state wavefunctions are normalised to $\langle b \mid b\rangle^2 = 1$.

Free state wavefunctions are normalised per unit Ry: $\langle E''f \mid E'f\rangle = \delta(E' - E'')$.

The reduced dipole matrix element is the sum of two contributions

$$D(a,b) = (a \,||\, \boldsymbol{D} \,||\, b) = D^{(\mathrm{I})} + D^{(\mathrm{O})} \tag{45}$$

$D^{(\mathrm{I})}$ from the internal region $r \leq a$ and $D^{(\mathrm{O})}$ from the external region $r \geq a$.

- $D^{(\mathrm{I})}$, the internal region contribution.

Let two physical states Ψ_a and Ψ_b have expansion coefficients A_{ak} and $A_{bk'}$ in Eq. (19). The internal region contribution to Eq. (45) is

$$\begin{aligned} D^{(\mathrm{I})}(a,b) &= (\Psi_a \,||\, D^{(\mathrm{I})} \,||\, \Psi_b) \\ &= \sum_{k,k'} A_{ak} M(k,k') A_{bk'} \end{aligned} \tag{46}$$

where

$$M(k,k') = (\psi_k \,||\, D^{(\mathrm{I})} \,||\, \psi_{k'}) \tag{47}$$

the round brackets as usual indicating the finite range of the radial integrations.

To evaluate this, let φ_λ denote collectively the basis functions and the $V_{k\lambda}$ denote collectively the coefficients c_{ijk} and d_{jk} in Eq. (12). Then define a reduced matrix $\boldsymbol{D}$ with elements

$$D_{\lambda\lambda'} = (\varphi_\lambda \,||\, D^{(\mathrm{I})} \,||\, \varphi_{\lambda'}) \tag{48}$$

Eq.(47) can thus be written in the following matrix form:

$$M(k,k') = \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{D} \boldsymbol{V}_{k'} \tag{49}$$

The A_{Ek} coefficients in Eq. (46) are given by Eq. (22), which, with $b = 0$ (its usual value in practice), can be written as

$$\begin{aligned} A_{Ek} &= \tfrac{1}{2}(E_k - E)^{-1} \sum_i w_{ik}(a) \left(\frac{\mathrm{d}F_i}{\mathrm{d}r}\right)_{r=a} \\ &= \tfrac{1}{2a}(E_k - E)^{-1} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{F} \end{aligned} \tag{50}$$

where we have used Eq. (23), and where the $\boldsymbol{w}$ and $\boldsymbol{R}$ matrices are given by Eq. (20) and Eq. (26). Defining diagonal matrices $\boldsymbol{G}_a$ and $\boldsymbol{G}_b$ with diagonal elements

$$G_{Ek} = \tfrac{1}{2a}(E_k - E)^{-1}$$

then Eq. (46) becomes

$$D^{(\mathrm{I})}(a,b) = \boldsymbol{F}_a^{*\mathrm{T}} \boldsymbol{R}_a^{-1} \boldsymbol{w}_a \boldsymbol{G}_a \; \boldsymbol{M} \; \boldsymbol{G}_b \boldsymbol{w}_b^{\mathrm{T}} \boldsymbol{R}_b^{-1} \boldsymbol{F}_b \tag{51}$$

The $\boldsymbol{F}$ matrices are given by Eq. (35) or Eq. (40) depending on whether free states or bound states are involved, and will be considered in more detail in the next subsections.

- $D^{(\mathrm{O})}$, the external region contribution.

In the external region antisymmetrisation can be neglected and, considering the case of the length operator, we can put

$$\boldsymbol{D} = \mathbf{R} + \boldsymbol{r}$$

where $\mathbf{R}$ is the operator for a transition in the target and $\boldsymbol{r}$ that for a transition by the outer electron. We then have

$$D^{(0)}(a,b) = \sum_{ii'} [\alpha_{ii'}(F_{ia} \mid F_{i'b}) + \beta_{ii'}(F_{ia} \mid r \mid F_{i'b})] \tag{52}$$

where

$$\alpha_{ii'} = (\overline{\Phi}_i \,||\, \mathbf{R} \,||\, \overline{\Phi}_{i'}), \qquad \beta_{ii'} = (\overline{\Phi}_i \,||\, \hat{\boldsymbol{r}} \,||\, \overline{\Phi}_{i'}) \tag{53}$$

and where $\overline{\Phi}_i$ are the functions in expansion Eq. (12).

The coefficient $\alpha_{ii'}$ is non-zero only if there is an optically-allowed transition between the target states belonging to channels i and i', and if $l_i = l_{i'}$.

The coefficient $\beta_{ii'}$ is purely algebraic and is non-zero only if the channels i and i' belong to the same target state, and if $l_i = (l_{i'} \pm 1)$.

The methods used to evaluate the integrals in Eq. (52) are described by Seaton [76].

1.3.3. Oscillator strengths

Having calculated the dipole matrix, we can now define the line-strength for a dipole transition between two states a and b of energy (in Ry) E_a and E_b in the length and velocity formulations:

$$\mathcal{S}_{\rm L}(b;a) = |\,(b \,||\, D_{\rm L} \,||\, a)\,|^2 \tag{54}$$
$$\mathcal{S}_{\rm V}(b;a) = 4\,(E_b - E_a)^{-2}\,|\,(b \,||\, D_{\rm V} \,||\, a)\,|^2$$

The oscillator strength is the dimensionless quantity $f(b,a)$ defined by

$$g_a f(b,a) = \tfrac{1}{3}(E_b - E_a)\mathcal{S}(b;a) \tag{55}$$

where g_a is the statistical weight or degeneracy of the initial state: $(2S_a+1)(2L_a+1)$ or $(2J_a+1)$, depending on the coupling scheme. The reduced dipole matrix elements in Eq. (54) are determined as in the previous subsection, using a wavefunction satisfying the bound-state asymptotic boundary conditions of Eq. (39).

Use of exact functions would give $\mathcal{S}_{\rm L} = \mathcal{S}_{\rm V}$. With approximate functions the differences between $\mathcal{S}_{\rm L}$ and $\mathcal{S}_{\rm V}$ give an indication of the accuracy achieved.

1.3.4. Photoionization cross section

Again, using the dipole matrix, we can define a generalised line-strength similar to Eq. (54) for a transition from an initial bound state i of energy E_i to a final free state f of energy $E_f = E_i + \omega$, where ω is the photon energy in Ry:

$$\mathcal{S}_{\rm L}(E_f;i) = \sum_{Ll_f} |\,(Ll_fE_f \,||\, D_{\rm L} \,||\, i)\,|^2 \tag{56}$$
$$\mathcal{S}_{\rm V}(E_f;i) = 4\,\omega^{-2}\sum_{Ll_f} |\,(Ll_fE_f \,||\, D_{\rm V} \,||\, i)\,|^2$$

The photoionization cross section is usually expressed (in either the dipole length form or the dipole velocity form) as

$$\sigma = \tfrac{4}{3}\pi^2 a_0^2 \alpha\,\tfrac{\omega}{g}\,\mathcal{S} \tag{57}$$

where g is the statistical weight of the initial bound state, and the generalised line-strength is calculated using final-state wavefunctions normalised per Ry. The constant in Eq. (57) is $\tfrac{4}{3}\pi^2 a_0^2\alpha = 2.689\ldots$Mb.

Let Ψ_i and $\Psi_f^-(\hat{\boldsymbol{k}})$ be the wavefunctions representing the initial bound state A_i of the target atom and the final scattering state ($e^- + A_f^+$) of the ejected electron plus residual ion. The boundary conditions satisfied by

Ψ_i correspond to decaying waves in all channels, while those satisfied by $\Psi_f^-(\hat{\boldsymbol{k}})$ correspond to a plane wave in the direction of the ejected electron momentum $\hat{\boldsymbol{k}}$ and ingoing waves in all open channels.

Both Ψ_i and $\Psi_f^-(\hat{\boldsymbol{k}})$ are now expanded in terms of the R-matrix basis in the internal region defined by Eq. (12). The appropriate expansions follow from Eq. (19) giving

$$\Psi_i = \sum_k \psi_k A_{ki} \quad \text{and} \quad \Psi_f^-(\hat{\boldsymbol{k}}) = \sum_k \psi_k A_{kf}^-(\hat{\boldsymbol{k}}) \tag{58}$$

The coefficients A_{ki} and $A_{kf}^-(\hat{\boldsymbol{k}})$ are determined by solving the differential equations in the external region, subject to the bound state and free state boundary conditions discussed above, and matching to the R-matrix boundary condition at $r = a$ as in Eq. (50). Note that the energy dependence is carried through the A_{Ek} coefficients; the energy independent ψ_k basis functions are the same in both the bound and the free state.

The generalised line strength used for photoionization calculations is expressed in terms of the reduced dipole matrix $D^{(1)}(f,i)$ defined by Eq. (51) using the functions

$$F_a^* \equiv (\boldsymbol{F}^-)^* = \boldsymbol{F}^+$$

for the final state, where

$$\boldsymbol{F}^- = -\mathrm{i}\boldsymbol{F}(1 - \mathrm{i}\boldsymbol{K})^{-1}$$

and the $\boldsymbol{F}(r)$ are functions with K-matrix asymptotic forms Eq. (31).

Thus, using R-matrix theory, we can calculate the photoionization cross section of Eq. (57) as well as other observables, such as the differential cross section and β asymmetry parameter (Burke and Robb [22]).

1.3.5. Other processes

The above discussion can be extended to enable free-free transitions to be calculated (Bell [10]). In this case both the initial and final states in the matrix elements in Eq. (44) are scattering states subject to appropriate ingoing and outgoing wave boundary conditions.

Allison [2] and Shorer [80] showed that the frequency dependent polarizability $\alpha(\omega)$ can be written in terms of an expansion in R-matrix basis functions. This follows immediately from the above theory using the well known result that the imaginary part of the polarizability $\alpha(\omega)$ is proportional to the total photoionization cross section. Shorer also extended this theory to enable two photon photoionization cross sections to be calculated.

The R-matrix defined by Eq. (24) is a meromorphic function of energy with poles occurring only on the real energy axis. The branch cuts in the S-matrix which occur at each of the thresholds arise from the solution of the problem in the external region. If Eq. (30) contains only the long-range Coulomb potential, one can use R-matrix theory to derive Seaton's multichannel quantum defect theory (MQDT) (Seaton [74]). This close link between R-matrix theory and multichannel quantum defect theory has been emphasised by Lane [57], and has been the basis of extensive calculations by Greene and Aymar [48] and collaborators who have combined MQDT and R-matrix techniques to describe the spectrum of the alkaline-earth atoms.

1.4. Relativistic R-matrix theory

As the charge Z on the nucleus increases, relativistic effects both in the target wavefunction and in the wavefunction representing the scattered electron become important even for low energy electron scattering.

For electrons with kinetic energies far below the rest energy $mc^2 = 511\,\text{keV}$ the Breit-Pauli Hamiltonian

$$H_{\text{BP}}^{N+1} = H^{N+1} + H_{\text{REL}}^{N+1}$$

discussed by Bethe and Salpeter [16] for the case of one- and two-electrons suffices as an equation of motion. H is the non-relativistic Hamiltonian given by Eq. (4). H_{REL} gives rise to perturbative contributions whose relative magnitudes are low powers of α.

The non-relativistic R-matrix method has been extended to include relativistic terms from the Breit-Pauli Hamiltonian by Scott and Burke [71]. In the current code we explicitly retain only the one-electron terms resulting from the reduction of the Dirac equation to Breit-Pauli form up to order $\alpha^2 Z^4$, i.e. the mass-correction term, the one-electron Darwin term and the spin-orbit term; implicitly accounted for are fine-structure two-electron contributions from closed subshells as outlined in Section 2 on STG1.

The low-Z Breit-Pauli Hamiltonian for an $(N+1)$-electron system is taken to be

$$H_{\mathrm{BP}}^{N+1} = H^{N+1} + H_{\mathrm{mass}}^{N+1} + H_{\mathrm{D}_1}^{N+1} + H_{\mathrm{SO}}^{N+1} \tag{59}$$

Each of the one-electron Breit-Pauli terms can optionally be included:

$$H_{\mathrm{mass}}^{N+1} = -\frac{\alpha^2}{8}\sum_{n=1}^{N+1}\nabla_n^4 \qquad \text{(mass-correction)} \tag{60}$$

$$H_{\mathrm{D}_1}^{N+1} = -\frac{\alpha^2 Z}{8}\sum_{n=1}^{N+1}\nabla_n^2\left(\frac{1}{r_n}\right) \qquad \text{(Darwin)} \tag{61}$$

$$H_{\mathrm{SO}}^{N+1} = \frac{\alpha^2 Z}{2}\sum_{n=1}^{N+1}\frac{\boldsymbol{l}_n \cdot \boldsymbol{s}_n}{r_n^3} \qquad \text{(spin-orbit)} \tag{62}$$

Note that the non-fine-structure part of the Hamiltonian

$$H_{\mathrm{nfs}}^{N+1} = H^{N+1} + H_{\mathrm{mass}}^{N+1} + H_{\mathrm{D}_1}^{N+1} \tag{63}$$

commutes with $\boldsymbol{L}^2$, $\boldsymbol{S}^2$, L_z, S_z and parity, whereas H_{SO}^{N+1} and H_{BP}^{N+1} only commute with $\boldsymbol{J}^2$, J_z and parity.

If the spin-orbit interaction is included, the $(N+1)$-electron R-matrix basis functions are defined as in Eq. (12), but for each total angular momentum J and parity. A pair-coupling scheme

$$\boldsymbol{J}_i + \boldsymbol{l} = \boldsymbol{K} \quad \text{and} \quad \boldsymbol{K} + \frac{\boldsymbol{1}}{\boldsymbol{2}} = \boldsymbol{J} \tag{64}$$

is used to evaluate the matrix elements (Racah [65], Scott and Burke [71]), where J_i is the total angular momentum of the target state, l the orbital angular momentum of the incident electron and $\frac{1}{2}$ is the spin of the incident electron.

In this approach, Hamiltonian matrices (and dipole matrices) are first calculated in LS-coupling, and then transformed using a unitary transformation to pair-coupling. This is described in more detail in Section 4 on RECUPD. The corresponding Hamiltonian matrix analogous to Eq. (13) in the non-relativistic case is now much larger, and there are more coupled channels in the external region Eq. (28), requiring a considerable increase in computational effort.

Providing the intermediate-coupling scheme of Eq. (64) is adhered to, the analysis of Sections 1.2 and 1.3 can be followed allowing electron scattering and photoionization to be computed with relativistic effects. This is implemented in RMATRX1.

1.5. Model potentials

In targets which contain a large number of electrons an appreciable saving of computational effort can be achieved by replacing the interaction of the valence and continuum electrons with the closed shell core by a model potential. This allows the calculation to proceed solely in terms of the valence and continuum electrons.

Consider the case of an N-electron atom. If there are N_c-electrons in the closed shell core then there are $M = N - N_c$ valence electrons. The Hamiltonian Eq. (4) becomes

$$H^{M+1} = \sum_{n=1}^{M+1} \left(-\tfrac{1}{2} \nabla_n^2 + V(r_n) + \sum_{m>n}^{M+1} \frac{1}{r_{nm}} \right)$$

where $V(r)$ is the model potential. Its specific form is left to the user with the proviso that it has the following limiting forms

$$V(r) \underset{r \to 0}{\sim} -\frac{Z}{r} \quad \text{and} \quad V(r) \underset{r \to \infty}{\sim} -\frac{z}{r}$$

where Z is the nuclear charge and $z = Z - N$ is the residual target charge. The model potential is read in from a file by module STG1 where it is stored numerically at a set of mesh points. If desired it may be angular momentum dependent. The basis states ψ_k of Eq. (12) are now $(M+1)$-electron functions where the Φ are built from the valence orbitals. However, it should be noted that for the purpose of orthogonalisation, in the determination of the continuum orbitals, the complete set of one-electron bound orbitals (including core functions) are required to be read into STG1.

1.6. Programming details

1.6.1. Structure of RMATRX1

The program modules are run sequentially as outlined in Fig. 1. Calculations can be in LS-coupling, or in intermediate-coupling using the optional route through RECUPD.

Disk files are produced at each stage, again outlined in Fig. 1; these intermediate files are normally deleted after use. The most useful output files are the H and D files produced at the end of STGH: these contain the diagonalized Hamiltonian matrices and any dipole matrices respectively, and contain all the data from the internal region required for the external region codes.

1.6.2. Comment on module STG4

The purpose of this write-up is to publish the internal-region modules, STG1, STG2, RECUPD, STGH and STGLIB. For completeness it was decided to include an external region module STG4 which provides for most of the observables. It is however based on the differential equation solver of Crees [37], which is not the fastest program available. It is however very general. For more specific purposes there are better programs, but for various reasons these cannot be published here. The user is therefore recommended to investigate the programs mentioned in Section 10 before starting serious production runs.

1.6.3. Language and precision

RMATRX1 is written in FORTRAN 77. It is designed to be transportable and has been implemented on a number of computers with comparative ease. The code can be preprocessed for dimensions. This has been kept simple and if necessary could be carried out with a text editor.

The programs use double precision. This is appropriate for Unix workstation, VAX, IBM and similar compilers.

Compilation on the Cray can be made using the –dp option.

VAX compilation should be made using the /G_FLOAT option to avoid overflows and underflows.

```
        CIV3/SUPERSTRUCTURE etc.
                 I
                 I
                 I
                STG1
                 I
                 I
        (RK.DAT) I (STG1.DAT)
                 I
                 I
             STG2+STGLIB
                 I
     (STG2H.DAT) I (STG2D.DAT)
                 I................
                 I               .
                 I        RECUPD+STGLIB
                 I               .
                 I   (RECUPH.DAT) (RECUPD.DAT)
                 I               .
                 I................
                 I
                STGH
                 I
                 I
         (H.DAT) I (Dnn.DAT)
                 I
                 I
             STG4+STGLIB etc.
```

Fig. 1. Structure of RMATRX1. Notation: vertical lines joining the modules indicate the sequential execution of the package, from top to bottom; a dotted line indicates the optional path through RECUPD. Note that module STGLIB is not an executable module on its own, it contains a library of routines. (Files used to transfer data from one module to another are indicated in brackets).

1.6.4. Before you start using RMATRX1

You will need a good target description. This can be obtained from an atomic structure package such as CIV3 [52] or SUPERSTRUCTURE [43].

Time spent at this stage is vital. There is a limit to the number of configurations that you can reasonably use with present day resources. Careful consideration of the target will save time later.

1.6.5. Using RMATRX1

You should always run through the programs with a small (a single electron+target symmetry) calculation first. This is just to gauge the size of the problem that you are attempting and check the input.

Before a large scale calculation :

1. Check that the *R*-matrix boundary is correct.
2. Check that the radial mesh is adequate. Be conservative and pick a mesh with a small step-size. The program does this automatically but you should still check. You should have sufficient mesh points (say 15) between the nodes of each orbital. It is a good idea to examine the overlap integrals in STG1 (NBUG5=1) and the bound-bound one-electron integrals (NBUG8=1).
3. Check that the target potential is correct. You might consider plotting the potential and the bound orbitals.
4. Check that the number of continuum orbitals per angular momentum (variable NRANG2) is adequate. It should be at least 20 for heavy atoms. This is partly determined by the energy range over which you wish to calculate cross-sections. Again this emphasizes the importance of a small preliminary calculation.
5. Check that the STG2 and RECUPD target energies are consistent with the target energies originally calculated by the atomic structure program.

6. Check that LAMAX allows for all the couplings that you wish to include.

1.6.6. I/O

Input data is read from unit number IREAD which is set to 5. The input file is opened internally and has the name `file.INP` where `file` would be STG1, STG2, RECUPD, STGH or STG4. If the input file does not exist then the program terminates with an error message.

Formatted output is written to unit number IWRITE which is set to 6. This output file is opened internally and has the name `file.OUT` where `file` would again be STG1, STG2, RECUPD, STGH or STG4. If a file with this name already exists then it is overwritten.

Most of the files used in the program are opened with STATUS set to UNKNOWN. This means that if the file already exists, then it will be overwritten. It is important that you rename files that you wish to keep.

The unit numbers and file names are set internally. Although the input data requests unit numbers these are generally not used. This redundancy in the input data has been retained to maintain input compatibility with earlier versions of the programs. Occasionally an input unit number is used as an option flag.

2. Module STG1

This module is the first stage in RMATRX1. It calculates the orbital basis and all radial integrals in the inner-region.

Figs. 2–5 displays a flow diagram for the routines in STG1.

There are three main computational sections in STG1 (the controlling routines are named in brackets):

- initialisation and reading input files (BLOCK DATA, STG1RD, ISTG1);
- evaluating the continuum orbitals (BASORB, WRITAP);
- calculating multipole, one- and two-electron radial integrals (GENINT).

Notation:

In the following description, we use the symbol $U_j(r)$ to denote any radial orbital from the set of bound orbitals:

$$P_{nl}(r) \quad \text{for} \quad n = l+1, \mathtt{MAXNHF}(l+1) \quad \text{and} \quad l+1 = 1, \mathtt{LRANG1}$$

and continuum orbitals

$$u_{ij}(r) \quad \text{for} \quad j = 1, \mathtt{NRANG2} \quad \text{and} \quad l_i+1 = 1, \mathtt{LRANG2}$$

(cf. Eq. (18)). Note that these orbitals have all absorbed a factor r in their definition.

All orbitals $U_j(r)$ are tabulated on the internal-region r mesh, $(0, \mathtt{RA})$, and stored in STG1 in the $\mathtt{UJ}(r,j)$ array; bound orbitals being in the first $j = 1, \mathtt{NBOUND}$ locations.

An address array provides the index to any bound or continuum orbital:

$$\mathtt{UJ}(r,j) \quad \text{for} \quad j = \mathtt{IPOS}(n, l+1)$$

2.1. Routines

MNSTG1
is the program routine and contains all COMMON blocks used in STG1. It sets the /MEMORY/ pointers, and calls AASTG1.

```
 1  MNSTG1 AASTG1 BASORB ----
13                GENINT ----
37                ISTG1  EVALUE ABNORM
38                              CORECT
39                       POTF
40                       TABORB
41                SHRIEK
42                STG1RD ----
52                WRITAP

1.1  .DATA.
```

Fig. 2. Calling tree for module STG1. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

```
 1                BASORB ABNORM
 2                       EVAL   CORECT
 3                       FINDER BASFUN DERFUN
 4                                     DEVGL  DERFUN
 5                                     MA01A
 6                              ROOT
 7                       NEWBUT BASFUN > 3
 8                              BUTFIT
 9                              LSQ    MA01A
10                       PHASE
11                       RECOV2
12                       SCHMDT
```

Fig. 3. Calling tree for module STG1, BASORB section.

```
13                GENINT DA2
14                       GEN1BB ONEELE
15                              RDAR
16                              RMASS
17                              SPNORB RS     DERINT CORECT
18                       GEN1BC ONEELE
19                              RDAR
20                              RMASS
21                              SPNORB RS     DERINT CORECT
22                       GEN1CC ONEELE
23                              RDAR
24                              RMASS
25                              SPNORB RS     DERINT CORECT
26                       GENBB  RS     DERINT CORECT
27                       GENBC  RS     DERINT CORECT
28                       GENCC  RS     DERINT CORECT
29                       GENMBB DERINT CORECT
30                              RADINT
31                       GENMBC DERINT CORECT
32                              RADINT
33                       GENMCC ABNORM
34                              DERINT CORECT
35                              RADINT
36                       RECOV2
```

Fig. 4. Calling tree for module STG1, GENINT section.

```
42            STG1RD MESH   RECOV2
43                   NAME   CIV3   CALORB CALEXO COEFF  MA01A
44                                                      ORNO
45                                        ORNO
46                          RECOV2
47                          SS     ABNORM
48                                 RECOV2
49                   RECOV2
50                   STO    ORNO
51                          RECOV2
```

Fig. 5. Calling tree for module STG1, STG1RD section.

AASTG1
is called by MNSTG1 and controls the STG1 computation, invoking the three main computational sections as summarised above.

ABNORM
calculates numerically the overlap integral

$$I = \int_0^{\mathrm{RA}} \mathrm{d}r\, U_i(r) U_j(r)$$

As in all the other routines involving numerical integration, Simpson's rule is used.

BASFUN
generates an R-matrix continuum function $u_{ij}(r)$, for r in the range $(0, \mathrm{RA})$, as the solution of the differential equations Eq. (15):

$$\left(\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{l(l+1)}{r^2} + V_0(r) + k^2\right) u(r) = \sum_n \Lambda_n P_{nl}(r) \qquad (65)$$

$$u(0) = 0$$

The logarithmic derivative on the boundary is

$$\left(\frac{\mathrm{RA}}{u(\mathrm{RA})}\right) \cdot \left(\frac{\mathrm{d}u}{\mathrm{d}r}\right)_{r=\mathrm{RA}} = b \equiv \mathrm{BSTO}$$

The potential $V_0(r)$ is input tabulated on the mesh and half-mesh points in array POVALU, as are the bound orbitals in array BNDORB. The continuum function is output on the mesh points in array ORB, and the Lagrange multipliers Λ_n, chosen to enforce orthogonality between the continuum and the bound orbitals as in Eq. (17), are output in ALAMDA.

There are two modes of operation, controlled by the argument ETA.

ETA> 10^{-8}: input the value of BSTO and the number of nodes to obtain a function satisfying logarithmic derivative boundary conditions, and return the eigenvalue k^2 to an accuracy of ETA (used in evaluating continuum functions);

ETA= 0: input the value of k^2 to obtain a function and its logarithmic derivative on the boundary (used in evaluating Buttle corrections).

BASFUN is a modified version of the program originally published by Robb [66] and modified by Berrington et al. [12,13]. De Vogelaere's integration method is used to solve the differential equations Eq. (65) at a set of mesh points also input to BASFUN. Both forward and backward integration is used with matching at some intermediate value of r. The eigenvalue is found by Newton's iteration method. The method appears to be stable for all but highly bound functions with large negative eigenvalues.

BASORB
is the controlling routine for the evaluation of the bound and continuum orbitals used in the R-matrix expansion of Eq. (12). Within the routine there is an outer loop over the angular momentum l = LP − 1, where LP = 1, max(LRANG1, LRANG2). The continuum orbitals are solutions of the differential equations Eq. (15) subject to logarithmic derivative boundary conditions as in Eq. (16), solved by routine BASFUN via a call to FINDER for each j = 1, NRANG2.

MAXNHF(LP), LP = 1, LRANG1 defines the maximum principal quantum number of all bound orbitals for l = LP − 1, and MAXNLG(LP) the maximum principal quantum number of the bound orbitals to which the continuum orbitals are to be Lagrange orthogonalised in the BASFUN package. BASFUN requires these latter bound orbitals to be tabulated in the P array at the radial mesh and half mesh points, and this is done by calling routines EVALUE or EVAL, for bound orbitals input to STG1 in Slater-type form or in numerical form respectively; routine EVALUE also stores the bound orbitals on the mesh points in the UJ array.

The summation over principal quantum numbers n in Eq. (15) normally goes over spectroscopic target orbitals occurring in the expansion of the atomic states for each angular momentum l_i = LP − 1. The Λ_{ijn} are Lagrange multipliers chosen to ensure orthogonalisation of the $u_{ij}(r)$ for all j with these MAXNLG(LP) bound orbitals as in Eq. (17). Any remaining bound orbitals (e.g. correlation orbitals) are Schmidt orthogonalised to the continuum basis for each l_i in routine SCHMDT.

The choice of BSTO $\equiv b$ in Eq. (16) is in principle arbitrary; normally BSTO = 0. In order to alleviate the constraint of a fixed slope at RA the diagonal elements of the R-matrix will be Buttle corrected, and a call to NEWBUT establishes the Buttle fit.

The continuum orbitals j for a given orbital value l are held in UJ(i,j)[4] in locations with $j >$ NBOUND, and their eigenvalues and boundary amplitudes are stored in the EIGENS and ENDS arrays.

For use in Breit-Pauli routines RDAR and RMASS further information is needed. At the origin we have

$$\lim_{r\to 0} u_j(r) = a_j r^{l+1} + b_j r^{l+2} + \dots \tag{66}$$

The Taylor coefficient a_j is stored in location UJ(1, j) after being transferred from routine BASFUN through ORB(1), as this array element would otherwise hold the trivial value 0.0 (the same convention applies to the first j = NBOUND functions UJ(i,j), which are used for storing $P_{n_j l_j}(r_i)$). Similarly the normalised Lagrange multipliers Λ_{ij} are passed through ALAMDA(I), I = 1, NBT and stored in array RLAMDA. The code makes use of RLAMDA when evaluating the one-electron integrals that arise from the mass-correction and one-electron Darwin terms.

In the case of neutral targets the function PHASE is called to calculate the zero order phase, which is printed out alongside each eigenvalue for l = 0, 1 and 2.

Options exist (NBUG5 ≥ 1) for writing out the tabulated orbitals, overlap integrals and any Schmidt coefficients.

BLOCK DATA
defines the data in common block /CONSTS/. This contains frequently used constants.

BUTFIT
is the default Buttle-correction routine, called by NEWBUT if BSTO = 0, parametrically fitting the Buttle correction of Eq. (25) over the energy range $E_{\min} \dots E_{\max}$ to solutions from a model Hamiltonian (Seaton [77]):

$$R^{c}(\text{NRANG2}, k^2) \approx \alpha B(\gamma, \beta + U) \tag{67}$$

[4] Here the index i runs over the radial mesh.

where α and β are the fitting parameters, and

$$B(M,U) = \tfrac{\tan(K)}{K} - 2\sum_{m=0}^{M}(U_m - U)^{-1}$$

where

$$\begin{aligned}U &= K^2\\ K &= ak\\ U_m &= K_m^2\\ K_m &= (m+\tfrac{1}{2})\pi\\ M &= \mathrm{int}(\tfrac{1}{2} + ak_{\mathrm{NRANG2}}/\pi)\end{aligned}$$

The fitting is done separately for each channel angular momenta l_i.

After returning to NEWBUT the coefficients COEFF(1,L) = α and COEFF(2,L) = β are stored, while the assignment COEFF(3,L) = $-1000 * \gamma$ serves as a convenient flag — to distinguish such data from LSQ output, as explained in the description of that routine.

CALEXO
is called from routine CALORB to read radial functions for excited and pseudo-orbitals from the program CIV3 [52] input data file; it returns orbital data in common block /RADIAL/.

CALORB
is called from routine CIV3 to read radial functions for Hartree-Fock orbitals from the program CIV3 [52] input data file; it returns orbital data in common block /RADIAL/.

CIV3
is called from NAME, but only if the user has specified to process target input in CIV3 [52] format rather than the standard (STO) input as in Eq. (70). This option is invoked by setting the first four characters on the first record of the input data file as 'CIV3'. CALORB and CALEXO are called to read Hartree-Fock and correlation orbitals.

COEFF
is called from CALEXO in the CIV3 input option, to determine the coefficients of the radial functions of a correlation orbital from orthonormality conditions.

CORECT
is called by EVALUE and also by DERINT when tabulating Slater-type orbitals and their derivatives. It evaluates the correction to be applied to each bound orbital to generate a new function

$$P(r) \to P(r) - c\,e^{-(r-r_0)^2/\sigma^2} \qquad (68)$$

This correction is chosen so that it is negligible everywhere except near the boundary, and that its value and logarithmic derivative cancel out the original orbital $P(r)$ at r = RA. The parameters r_0, σ and c are calculated from the amplitude and logarithmic derivative of the original orbital on the boundary, and from the condition that the Gaussian falls to $\exp(-s^2)$ of its maximum value at $r = 0.9$ RA. The quantity s is given the value 1.5 in CORECT.

DA2
is used to write output file 'RK.DAT' (on unit number JDISC1) for use in module STG2.

This routine is called to open, write, or read a binary direct access file for both temporary and permanent storage of the contents of a large array. The record length is data-independent and fixed at LREC words. Because a given call may read or write an array spanning more than one record in the file, the argument IREC is input to specify the starting record number for the current call; IREC is then incremented to return the next available record number.

Routine DA2 assumes a word length LWORD = 8; this PARAMETER may be reduced in the code when running the program on machines with shorter word length if disk resources are scarce; it must be increased if longer words are chosen.

DERFUN
is called by BASFUN. It evaluates the second derivative of $u(r)$.

DERINT
evaluates the dipole velocity radial integral

$$I_V(i,j) = \int_0^{\mathrm{RA}} \mathrm{d}r\, U_i(r) \left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{\alpha}{r}\right) U_j(r) \tag{69}$$

using Simpson's rule and

$$\alpha = \begin{cases} l_j & \text{if } l_i = l_j - 1 \\ -l_j - 1 & \text{if } l_i = l_j + 1 \end{cases}$$

If both $U_i(r)$ and $U_j(r)$ are continuum orbitals the first derivative is evaluated numerically. Otherwise it is evaluated analytically by reversing the orbitals if necessary (and the sign of I_V) to ensure the right-hand orbital $U_{n_j l_j}(r)$ is bound.

DEVGL
is called by BASFUN. It propagates solutions $u(r)$ one step using de Vogelaere's method.

EVAL
is a variant of EVALUE for dealing with numerically supplied orbitals held in UJ. It evaluates the midpoint values of user-supplied NBOUND target orbitals and stores them in P of /BNDORB/.

EVALUE
is called by BASORB and evaluates bound functions from Slater-type orbital (STO) parameters,

$$\mathrm{UJ}(I,K) = P_{nl}(r) = \sum_{J=1}^{\mathrm{NCO}} \mathrm{C}(J) * r^{\mathrm{IRAD}(J)} * \mathrm{e}^{-\mathrm{ZE}(J)*r} \tag{70}$$

where K = IPOS$(n, l+1)$ $\leq$ NBOUND. The functions are tabulated at radial points $r \equiv$ XR(I) between XR(2) = HINT and XR(NPTS) = RA. The IPOS address array is also defined. As in the context of continuum functions as in Eq. (66) it is convenient to assign

$$\mathrm{UJ}(1,K) = \lim_{r\to 0} \frac{P_{nl}(r)}{r^{l+1}} \tag{71}$$

It is also convenient to store

$$\mathrm{DUJ}(I,K) = Q_{nl}(r) \equiv \left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} + \frac{l(l+1)}{r^2} - \frac{2Z}{r}\right) P_{nl}(r) \tag{72}$$

readily derived analytically from Eq. (70), for radial points r_I. In DUJ$(1,K)$ special quantities are held for RMASS. The tail of STOs and their radial derivatives is modified according to CORECT.

In addition spectroscopic orbitals in the range $I = 2,\mathtt{NPTS}$ and at the $2 * \mathtt{NPTS} - 2$ midpoints are stored in `P` for use by routine BASFUN for a particular value of orbital l at a time.

FINDER
is called by BASORB and in turn calls BASFUN. It selects solutions with the required number of nodes. Routine ROOT is called to improve the estimate of the continuum orbital energy.

GEN1BB
controls the computation of all bound-bound one-electron integrals by calling ONEELE, i.e. with bound orbitals $P_{n_1 l_1}$ and $P_{n_2 l_2}$; also calls RMASS, RDAR and SPNORB for Breit-Pauli corrections. The outer loop is over angular momentum ($l_1 = l_2$ for a nonzero integral) Note that because of the symmetry condition on the principal quantum numbers, only those integrals for which $n_2 \leq n_1$ are evaluated. The loops are arranged so that n_2 varies the most rapidly, so the lower triangle of the matrix of integrals defined by $(n_1\ , n_2)$ are stored consecutively by rows in array `ONEST1` with pointers in array `IST1` for each angular momentum.

GEN1BC
is the same as GEN1BB but in this case one radial function is of type ‘continuum’. In this case no symmetry relations can be exploited to reduce their number; the matrix of integrals are stored consecutively by rows in array `ONEST2` with pointers in array `IST2` for each angular momentum.

GEN1CC
is the same as GEN1BB but in this case both radial functions are of type ‘continuum’, i.e. involving continuum orbitals u_{ij} and $u_{i'j'}$. GEN1CC is called separately for each angular momentum. Again because of the symmetry condition, only those integrals for which $j' \leq j$ are evaluated, so the lower triangle of the matrix of integrals are stored consecutively by rows in array `ONEST3`.

GENBB
controls the computation of two-electron integrals when all four radial functions are of type ‘bound’. Routine RS is called. These integrals are defined by

$$R^k(n_1l_1, n_2l_2, n_3l_3, n_4l_4) = \int_0^{\mathrm{RA}} \mathrm{d}r \int_0^{\mathrm{RA}} \mathrm{d}s\, U_{n_1l_1}(r) U_{n_2l_2}(s) \frac{r_<^k}{r_>^{k+1}} U_{n_3l_3}(r) U_{n_4l_4}(s) \tag{73}$$

where $r_<$ and $r_>$ are the lesser and greater of r and s respectively. The following symmetry properties, imposed on the quantum numbers in the integral, allow a considerable saving in the number of integrals evaluated:

$$| l_1 - l_3 | \leq k \leq l_1 + l_3 \tag{74}$$

$$| l_2 - l_4 | \leq k \leq l_2 + l_4 \tag{75}$$

$$l_1 + l_2 + l_3 + l_4 = \text{an even integer} \tag{76}$$

$l_1 \leq l_2,\ l_1 \leq l_3,\ l_2 \leq l_4$

if $l_1 = l_2$ then $n_1 \geq n_2$

if $l_1 = l_3$ then $n_1 \geq n_3$

if $l_1 = l_4$ then $n_2 \geq n_4$

The loops are arranged so that n_4 varies most rapidly, n_3 varies next most rapidly, etc. over their allowed ranges. Thus the integrals defined by $(n_1 n_2 n_3 n_4)$ are stored consecutively by rows in array `RKST01`, with pointers to the $(l_1l_2l_3l_4)$ combinations stored in array ICTBB, and pointers corresponding to the k in array ISTBB2 (the k values are stored in array ISTBB1).

GENBC
controls the computation of two-electron integrals for which one radial function is of type 'continuum'. Routine RS is called. In this case the quantum numbers n_4l_4 in Eq. (73) correspond to the continuum function, while the other nl quantum numbers correspond to bound functions. Symmetry conditions in Eqs.(74–76) apply, but because the continuum orbital must always remain in the position n_4l_4, only the following further conditions apply:

$l_1 \le l_3$

if $l_1 = l_3$ then $n_1 \ge n_3$

The loops are arranged so that n_4 varies most rapidly, n_3 varies next most rapidly, etc. over their allowed ranges. Thus the integrals defined by $(n_1n_2n_3n_4)$ are stored consecutively by rows in array RKST02, with pointers to the $(l_1l_2l_3l_4)$ combinations stored in array ICTBC, and pointers corresponding to the k in array ISTBC2 (the k values are stored in array ISTBC).

GENCC
controls the computation of two-electron integrals for which two radial functions are of type 'continuum'. Routine RS is called. GENCC is called for specific continuum angular momenta L and LP. There are two distinct cases of direct and exchange integrals to be considered.

- Direct integrals.
  The continuum functions always have the positions n_2l_2 and n_4l_4 in Eq. (73), i.e. L = l_2 and LP = l_4. Symmetry conditions in Eqs.(74–76) apply, with the following further conditions:

  $l_1 \le l_3$, $l_2 \le l_4$

  if $l_1 = l_3$ then $n_1 \ge n_3$

  if $l_1 = l_4$ then $n_2 \ge n_4$

  The loops are arranged so that n_4 varies most rapidly, so that the integrals defined by $(n_1n_2n_3n_4)$ are stored consecutively by rows in array RKST02, with pointers to the (l_1l_3k) combinations stored in array ICTCCD.
- Exchange integrals.
  The continuum functions always have the positions n_2l_2 and n_3l_3 in Eq. (73), i.e. L = l_2 and LP = l_3. Symmetry conditions in Eqs.(74–76) apply, with the following further conditions:

  $l_2 \le l_3$

  if $l_2 = l_3$ then $n_2 \ge n_3$

  The loops are arranged so that n_3 varies most rapidly, so that the integrals defined by $(n_1n_4n_2n_3)$ are stored consecutively by rows in array RKST02, with pointers to the (l_1l_4k) combinations stored in array ICTCCE.
  A problem arises for high k (> 20) when the integral increases unexpectedly with k. This is due to inherent instability in the algorithm in routine RS (i.e. , it is not affected by the radial mesh). In practice this only affects exchange integrals at high l which should vanish anyway. A test is made for this problem and if it is detected the exchange integrals are set to zero.

There is a facility in GENCC for dealing with array overflow of the RKST02 array. If this array gets full, control is returned to the calling routine GENINT which dumps the contents of the array onto the output file 'RK.DAT'; the variable IRK2, which is the number of integrals in RKST02, is set negative and stored as JRK2 on output file ITAPE3. GENCC is then re-called to continue integral evaluation, re-using the RKST02 array.

GENINT
is in overall control of the computation of the multipole, one- and two-electron radial integrals.

GENMBB
generates all the bound-bound multipole length and dipole velocity integrals including Buttle correction type dipole integrals. For the dipole integrals involving a given pair of orbitals, routines RADINT and DERINT are called consecutively to calculate the length and velocity forms respectively; for higher multipoles, only RADINT is called for the length form.

GENMBC
is the same as GENMBB but in this case one radial function is of type 'continuum'.

GENMCC
is the same as GENMBB but in this case both radial functions are of type 'continuum'.

ISTG1
is called once from AASTG1 for initialisation.

- Defines mesh parameters: the radial mesh points (array XR) and the number of mesh points (NPTS). Also defines step-lengths (array STEP), and Simpson's rule weights (array WT), such that a finite integral over an arbitrary function $f(r)$ can be evaluated numerically as a weighted sum:

$$\int_0^{\mathrm{RA}} \mathrm{d}r\, f(r) \approx \sum_{I=1}^{\mathrm{NPTS}} \mathtt{WT}(I) * f(\mathtt{XR}(I))$$

- Defines powers of the radial mesh points required for many of the integral evaluations:

$$\mathtt{RK}(I,K) = \mathtt{XR}(I) * *K = r^K$$

for integer K in the range $(-\mathtt{L12}-2, \mathtt{L12})$ where

$$\mathtt{L12} = \mathtt{LRANG1} + \max(\mathtt{LRANG1}, \mathtt{LRANG2}) - 2$$

- Tabulates the analytic bound orbitals $P_{nl}(r)$ on the radial mesh by calling EVALUE.
- Tabulates the zero-order potential $V_0(r)$ by calling POTF.

LSQ
is called by NEWBUT if BSTO $\neq 0$. It matches the Buttle amplitude in a least squares quadratic fit for each continuum orbital $l = L - 1$ and returns the 3 coefficients c_i for storing in COEFF(i, L). Since the curvature coefficient c_3 is small and positive, LSQ returns a result that can be readily distinguished from that of BUTFIT. The method used in BUTFIT is preferred.

MA01A
is a linear equations solver. It is called by BASFUN, COEFF and LSQ.

MESH
generates the internal region radial mesh parameters NIX, IRX, IHX and HINT in /INIT/ as a function of Z and NRANG2. The mesh is used to tabulate all bound and continuum radial orbitals and is suitable for Simpson's rule integrations.

The mesh $r = 0$ to $r = \mathrm{RA}$ is divided into a small number (NIX) of intervals; the mesh points are equally spaced within each interval. The mesh is finest near $r = 0$. Using the criterion that the $1s$ bound orbital is the most rapidly varying near the origin, the first interval has a step length

$$h_{\min} = \mathtt{HINT} \approx 0.025/Z$$

The mesh is coarsest near r = RA and is taken to provide at least 16 mesh points between nodes. Using the criterion that the highest continuum orbital is the most rapidly varying near the boundary, the final interval has a step length

$$h_{\max} = \mathtt{HINT} \times \mathtt{IHX(NIX)} \approx \mathtt{RA/NCORSE}$$

where NCORSE = NRANG2 × 16 normally, since the separation between nodes of the highest continuum orbital tends to π/k_{NRANG2} with $k_{\mathrm{NRANG2}} = \pi \times$ NRANG2/RA, assuming a $\sin kr$ behaviour and a zero derivative condition at r = RA. However, in the case of large angular momentum l, the continuum orbital nodes are more tightly packed towards the boundary and the eigenvalue increases with $k_{\mathrm{NRANG2}} \approx \pi(\mathrm{NRANG2} + l/2)/\mathrm{RA}$, assuming a $\sin(kr - l\pi/2)$ behaviour, so NCORSE is modified accordingly for large LRANG2.

Each interval is made up of 16 steps or a small multiple thereof, and the step length IHX doubles from one interval to the next:

$$\mathtt{RA} = \sum_{I=1}^{\mathtt{NIX}} N_I * \mathtt{HINT} * \mathtt{IHX}(I) \tag{77}$$

where the number of points in the Ith interval is N_I = IRX(I) − IRX(I − 1) with N_1 = IRX(1); and where IHX(I) = 2 ∗ IHX(I − 1) with IHX(1) = 1.

The program aborts in MESH if this procedure leads to more than &NPT tabulation points. In this case one may reprocess — or run the program with a reduced number NRANG2 of continuum orbitals if this is satisfactory.

MESH is not called if the user supplies radial orbitals in option TITLE(1) ='S.S.'; the radial grid supplied alongside is used instead, in expanded form as described in routine SS if necessary.

NAME
is called by STG1RD if TITLE(1) equals 'CIV3', 'S.S.' or 'STO-'. After reading at least the 3 compulsory input parameters in NAMELIST format it calls CIV3, SS or returns, according to the value of TITLE(1), so as to read and process user-supplied target orbital input.

NEWBUT
controls the fit of the Buttle correction of Eq. (25) over an energy range between EK2MIN $\approx -Z^2/n^2$ (where n is the smallest principal quantum number likely to occur for a given value NELC) and the highest collision energy EK2MAX, which is conventionally set to $\approx k^2_{\mathrm{NRANG2}}/2$. While steering clear of poles it selects up to IMAX = 6 fairly equidistantly spaced points for each continuum orbital set l and parametrises the Buttle correction, as discussed in Seaton [77]. If BSTO is exactly zero NEWBUT calls BUTFIT, otherwise LSQ, and it stores the resulting 3 parameters of the energy fit in array COEFF for each l. A message is printed if EK2MAX is too small to support at least 3 function values for interpolation.

ONEELE
computes the non-relativistic one-electron integrals for all combinations of bound and continuum functions using Simpson's rule. Note that the integral is zero unless the angular momenta of the two functions involved are equal. Also the integral is unchanged on reversal of the two functions. The routine employs a number of strategies to avoid calculating second derivatives numerically.

- Bound-bound and bound-continuum integrals.

For the non-relativistic one-electron operator

$$h = -\frac{\mathrm{d}^2}{\mathrm{d}r^2} + \frac{l(l+1)}{r^2} - \frac{2Z}{r}$$

we note that bound orbital functions $Q_j(r_i) = hP_j(r_i)$, defined in Eq. (72), have been written into DUJ(i,k) ($k \leq$ NBOUND) by EVALUE or SS, so that if there is at least one bound orbital P_{nl} involved in the integral:

$$Q(m,n) = \tfrac{1}{2} \int_0^{\mathrm{RA}} \mathrm{d}r\, U_m(r)\, h\, P_{nl}(r) \tag{78}$$

$$= \tfrac{1}{2} \sum_{I=2}^{\mathrm{NPTS}} \mathtt{WT}(I) * \mathtt{UJ}(I,m) * \mathtt{DUJ}(I,n)$$

where $\mathtt{WT}(I)$ are the Simpson weights at radial mesh points r_I.

- Continuum-continuum integrals.

If both orbitals are of type continuum a different procedure is adopted, evaluating $hu_{ij}(r)$ from the knowledge that $u_{ij}(r)$ satisfies the second-order differential equations Eq. (15).

Let $M = \mathtt{MAXNHF}(l_i + 1) - \mathtt{MAXNLG}(l_i + 1)$.

For M=0 we have Lagrange orthogonalisation to all bound orbitals and Eq. (78) can be rewritten using Eq. (15):

$$Q(i,j) = -\tfrac{1}{2} \int_0^{\mathrm{RA}} \mathrm{d}r\, u_i(r) \left[\frac{2Z}{r} - V_0(r)\right] u_j(r) - \tfrac{1}{2} k_i^2 \delta_{ij}$$

k^2 is available in array EIGENS from routine BASORB and $[2Z/r - V_0(r)]$ in PX from POTF or SS (or is calculated from POTHAM and POVALU in the case of a user-supplied core potential, when IPSEUD $\neq 0$).

For $M > 0$ we have Schmidt orthogonalisation to M bound orbitals. It is then the new set of functions u_j'' generated in the SCHMDT routine that appear in Eq. (78); using Eq. (15):

$$Q(i,j) = -\tfrac{1}{2} \int_0^{\mathrm{RA}} \mathrm{d}r\, u_i''(r) \left[\frac{2Z}{r} - V_0(r)\right] u_j''(r) + \tfrac{1}{2} \sum_{J=M+1}^{\min(i,j)} B_{jJ} B_{iJ} k_J^2$$

$$+ \tfrac{1}{2} \sum_{J=M+1}^{j} \sum_{K=1}^{M} B_{jK} B_{jK} k_J^2 \int_0^{\mathrm{RA}} \mathrm{d}r\, P_{Kl}(r) u_K(r)$$

$$+ \tfrac{1}{2} \sum_{J=1}^{M} B_{jJ} \int_0^{\mathrm{RA}} \mathrm{d}r\, u_i''(r) \left(h + \left[\frac{2Z}{r} - V_0(r)\right]\right) P_{Jl}(r)$$

where the Schmidt coefficient matrix $\boldsymbol{B}$ has been evaluated in routine SCHMDT, and the overlap contributions of bound orbital functions $P_{nl}(r)$ with the continuum functions $u_j(r)$ before Schmidt orthogonalisation have been stored in array OVRLAP.

ORNO

is a function routine. It analytically evaluates the overlap integral of a pair of STOs.

PHASE

evaluates zero-order s-, p- and d-wave phases for electron-atom scattering.

POTF

calculates the zero-order potential function $V_0(r)$ used in Eq. (65) for evaluating continuum functions and stores it in POVALU. Its operation mode is controlled by NPOT. There are three modes of operation.

NPOT= 0:

this is the default if STOs have been specified for the target orbitals. POVALU = $V_0(r)$ is then calculated as the static central potential of an N-electron ground state from such orbitals. An array PX = $[2Z/r - V_0(r)]$ is also defined, for use in the one-electron routine ONEELE.

NPOT> 0:

user-specified potential, with data supplied according to Section 2.3:

$$V_0(r) = \sum_{I=1}^{\mathtt{NPOT}} \mathtt{CPOT}(I) * r^{\mathtt{IPOT}(I)} * \mathrm{e}^{-\mathtt{XPOT}(I)*r} \tag{79}$$

If numerical orbitals rather than STO coefficients are used then NAME sets NPOT = 2 and POTF is called with parameters such that

$$V_0(r) = \frac{2N\mathrm{e}^{-\sqrt[3]{Z}r}}{r} + \frac{2(Z-N)}{r} \tag{80}$$

For a neutral atom the equation obviously reduces to the case of NPOT = 1.

We note in passing that input generated by routine TABORB (NPOT = 0) in a previous calculation will conveniently lead to Eq. (80) both with NPOT = 0 and -1.

NPOT< 0:

along with S.S. type numerical input one may specify NPOT = -1, when POTF yields

$$V_0(r) = \frac{2}{r}\left\{\left(\mathcal{Z}_{\mathrm{eff}}^{\mathrm{SM}}(r) - Z\right) \cdot \frac{N}{N-1} + Z\right\} \tag{81}$$

where $\mathcal{Z}_{\mathrm{eff}}^{\mathrm{SM}}(r)$ is the effective $(N-1)$-electron charge read by SS; rescaling according to Eq. (81) is modelled on the behaviour of Eq. (80). The option Eq. (81) is very appealing for medium to heavy positive ions.

RADINT

computes radial multipole integrals of order $k \le \mathtt{LAM}$ for radiative transitions:

$$\begin{aligned} I_{\mathrm{L}}^{k}(i,j) &= \int_0^{\mathtt{RA}} \mathrm{d}r\, U_i(r)\, r^k\, U_j(r) \\ &= \sum_{I=2}^{\mathtt{NPTS}} \mathtt{WT}(I) * \mathtt{UJ}(I,i) * \mathtt{UJ}(I,j) * \mathtt{XR}(I) * *k \end{aligned} \tag{82}$$

RDAR

evaluates Darwin term radial coefficients if IRELOP(2) = 1; such a contact contribution to the one-electron Breit-Pauli interaction arises for pairs of s-orbital functions (i.e. $l_i = l_{i'} = 0$):

$$\begin{aligned} I_{\mathrm{D_1}}(i,i') &= -\frac{Z\alpha^2}{8}\left\langle i \left| \nabla^2\left(\frac{1}{r}\right) \right| i' \right\rangle \\ &= \frac{Z\alpha^2}{8} \lim_{r\to 0}\frac{U_i(r)}{r} \cdot \lim_{r\to 0}\frac{U_{i'}(r)}{r} \\ &= \frac{Z\alpha^2}{8}\mathtt{UJ}(1,K) * \mathtt{UJ}(1,K') \end{aligned} \tag{83}$$

where K = IPOS(i, 1) and K' = IPOS(i', 1).

RECOV2
is called from various routines when encountering insufficient array space. A message is printed enabling the user to recompile with larger preprocessor parameters. Abortion is deferred if more such checks can be made. This depends on the value of the input parameter IPLACE.

RMASS
computes radial integrals associated with the mass-correction Breit-Pauli term if IRELOP(1) = 1. The radial integral is

$$I_{\text{mass}}(j,j') = -\frac{\alpha^2}{8}\int_0^{\text{RA}} \mathrm{d}r\, U_j(r)\,\nabla^4\, U_{j'}(r) \tag{84}$$

$$= -\tfrac{\alpha^2}{8}\int_0^{\text{RA}} \mathrm{d}r \left(\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{l_j(l_j+1)}{r^2}\right) u_j(r) \left(\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{l_{j'}(l_{j'}+1)}{r^2}\right) u_j'(r)$$

Because the operator is purely radial such integrals only arise if $l_{j'} = l_j$ $(=l)$. Expression Eq. (84) is valid for bound and continuum orbitals $U(r)$. For bound orbitals, the second derivative is available in array UDP; for continuum orbitals, the second derivative is available from Eq. (65), i.e.

$$\left(\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{l(l+1)}{r^2}\right) u_j(r) = -\left(V_0(r) + k_j^2\right) u_j(r) + \sum_n A_n P_{nl}(r)$$

The integrand does *not* vanish at $r = 0$ if $l = 0$, but it is readily computed. For STOs the quantity $\left(\frac{\mathrm{d}^2 U_j(r)}{\mathrm{d}r^2}\right)_{r\to 0}$ can be evaluated analytically from Eq. (70), giving

$$\lim_{r\to 0}\frac{\mathrm{d}^2 P_{ns}(r)}{\mathrm{d}r^2} = 2\left\{\sum_{\text{IRAD}(J)=2} \text{C}(J) - \sum_{\text{IRAD}(J)=1} \text{C}(J) * \text{ZE}(J)\right\}$$

where the summations run over those STO terms in Eq. (70) containing powers of r and r^2 respectively. This is supplied by EVALUE through DUJ(1, K), where K is the bound orbital index IPOS($n, l+1$).

All the other types of orbital functions $U_j(r) \equiv \bar{U}_j(r) r^{l+1}$ which we consider satisfy the cusp condition

$$\lim_{r\to 0} U_j(r) = r^{l+1}\left\{\bar{U}_j(0)\left(1 - \frac{Zr}{l+1}\right) + O(r^2)\right\}$$

so that

$$\lim_{r\to 0}\frac{\mathrm{d}^2 U_j(r)}{\mathrm{d}r^2} = -2 * \text{Z} * \text{UJ}(1,K) \quad \text{if } l = 0$$

Also UJ(1, K) = $\bar{U}_j(0)$, see Eq. (66) and Eq. (71).

ROOT
is called by FINDER. It is an efficient algorithm for obtaining the zero of a non-linear equation (Shampine and Gordon [79]).

RS
has two modes of operation:

MODE= 0 (called from routines GENBB, GENBC, GENCC).

Computes the two-electron integral of Eq. (73) over four radial orbitals for given multipolarity k:

$$R^k(abcd) = \int_0^{\mathrm{RA}} \mathrm{d}r \int_0^{\mathrm{RA}} \mathrm{d}s\, U_a(r)U_b(s)\, \frac{r_<^k}{r_>^{k+1}}\, U_c(r)U_d(s)$$

$$= \int_0^{\mathrm{RA}} \mathrm{d}s\, U_b(s)\, y^k(ac;s)\, U_d(s)$$

with the multipole potential functions given by

$$y^k(ac;s) = \frac{1}{s^{k+1}} \int_0^{s} \mathrm{d}r\, U_a(r) r^k U_c(r) + s^k \int_s^{\mathrm{RA}} \mathrm{d}r\, U_a(r) \frac{1}{r^{k+1}} U_c(r)$$

We save the current $y^k(ac;s)$ in an array YK of /YSTOR/, because the looped calls to RS are suitably arranged so that this array can be re-used.

MODE= $+1$ or -1 (called from routine SPNORB).

Computes magnetic integrals of type N^k and V^k respectively:

$$N^k(abcd) = \int_0^{\mathrm{RA}} \mathrm{d}r\, r^2 \int_0^{\mathrm{RA}} \mathrm{d}s\, s^2\, R_a(r)R_b(s)\, \frac{r_<^k}{r_>^{k+3}}\, R_c(r)R_d(s)\, \epsilon(r-s) \tag{85}$$

$$V^k(abcd) = \int_0^{\mathrm{RA}} \mathrm{d}r\, r^2 \int_0^{\mathrm{RA}} \mathrm{d}s\, s^2\, R_a(r)R_b(s)\, \frac{r_<^k}{r_>^{k+3}}\, s\frac{\mathrm{d}}{\mathrm{d}r} R_c(r)R_d(s)$$

with the usual conventions; in particular $\epsilon(r-s)$ is the Heaviside step function and R a pure radial function — unlike U, which has absorbed a factor r from the volume element $r^2\mathrm{d}r$. The derivative

$$\frac{\mathrm{d}R}{\mathrm{d}r} = \frac{1}{r}\frac{\mathrm{d}U}{\mathrm{d}r} - \frac{U}{r^2}$$

is obtained on calling DERINT with argument $n_i = 0$, when it returns $\frac{\mathrm{d}U_i}{\mathrm{d}r}$ in YK.

RS (with MODE = 0) is the routine in which much of the STG1 computer time is spent.

SCHMDT

orthogonalises continuum orbitals $u_{ij}(r)$ with respect to those target orbitals $P_{nl_i}(r)$ for which $n >$ MAXNLG($l+1$), i.e. those target orbitals not included in the Lagrange orthogonalisation in BASFUN.

Let $M =$ MAXNHF($l+1$) $-$ MAXNLG($l+1$).

A new set of mutually orthogonal functions $u_j''(r)$, $j = 1, M +$ NRANG2, is generated from the set of bound $P_j(r)$, $j = 1, M$ and continuum $u_j(r)$, $j = 1,$ NRANG2 orbitals as follows:

$$u_m''(r) = P_m(r) \qquad m = 1, M$$

$$u_m''(r) = \sum_{j=1}^{M} B_{mj}P_j(r) + \sum_{j=M+1}^{m} B_{mj}u_j(r) \qquad m = M+1, M+\mathrm{NRANG2}$$

The Schmidt coefficients B_{mj} and the new functions u_m'' are generated as follows, for $m = M+1, M+$ NRANG2:

$$
\begin{aligned}
u'_m &= \sum_{j=1}^{m-1} \left[-(u_m \mid u''_j)\right] u''_j + u_m \\
N_m &= (u'_m \mid u'_m)^{1/2} \\
B_{mj} &= N_m \sum_{k=1}^{m-1} \left[-(u''_k \mid u_m)\right] B_{kj} \quad j = 1, m-1 \\
B_{mm} &= N_m \\
u''_m &= u'_m N_m
\end{aligned}
$$

where round brackets indicate the finite integral from $r = 0$ to RA. It is these $u''_j(r)$ functions which are stored in the UJ array and used in all the integration routines in STG1. The Schmidt coefficients B_{mj} are stored for later use in evaluating the one-electron integrals in routine ONEELE.

Numerical instability due to potential overcompleteness of the basis set can occur if NRANG2 is too large. It is therefore sensible to check for signs of numerical instability by looking at the debug printout of the overlap integrals $(u''_j(r) \mid u''_m(r))$ (which should resemble a unit matrix) and the Schmidt coefficients B_{mj} (which should not get too big). If the program thinks NRANG2 is too large, SCHMDT aborts returning a reduced value for NRANG2 that might enable the calling routine BASORB to restart the case more successfully — after printing a message.

SHRIEK
computes the factorial $n! = \mathtt{GAMMA}(n+1)$ in common block /FACT/.

SPNORB
is called if IRELOP(3) = 1. It computes the spin-orbit parameter

$$\zeta^0_{il,jl} = \alpha^2 \int_0^{\mathrm{RA}} \mathrm{d}r\, U_i(r) \frac{Z}{r^3} U_j(r)$$

or rather the integral

$$I_{\mathrm{SO}}(i,j) = \frac{Z\alpha^2}{2} \sum_{I=2}^{\mathrm{NPTS}} \mathtt{WT}(I) * \mathtt{UJ}(I,i) * \mathtt{UJ}(I,j) \,/\, \mathtt{XR}(I) ** 3 \tag{86}$$

for bound and continuum orbitals $l > 0$.

The program computes closed-shell effects on ζ associated with both outer target and continuum orbitals, unless told not to by setting IZESP > 0 (user supplies factors ZESP(l)) or IZESP $= -1$ (no screening). The interaction of an electron outside a closed shell nl through mutual spin-orbit effects behaves like an effective one-electron spin-orbit potential, reducing the 'naked' ζ^0. As no specific configuration $C = \prod_\iota (n_\iota l_\iota)^{q_\iota}$ can be considered at this stage it follows that not all the mutual spin-orbit terms with this behaviour according to Blume and Watson [18] are included in the procedure, rather only those common to all configurations for a given core. Thus if IZESP = 0 and NELC > 2 then screening of ζ^0 by $1s^2$ — and by L-, M- etc. shells if NELC is sufficiently large — is accounted for. Routine RS is called with argument MODE = ± 1 to obtain the magnetic integrals V^k and N^k of Eq. (85), and the code follows Jones [55] for the associated algebraic coefficients in the summations over closed-shell electrons and — in the exchange portion — multipolarities k.

Blume-Watson screened effective spin-orbit parameters ζ should also be computed when pseudo-potentials are supplied (option IPSEUD = +1), as inner shell orbital functions are normally supplied alongside POTHAM through routine SS. But an option IPSEUD = -1 has been implemented to compute the parameters ζ for both valence and continuum orbitals as expectation values of the operator

$$\frac{1}{r}\frac{\mathrm{d}V}{\mathrm{d}r} = \frac{\mathcal{Z}(r)}{r^3} - \frac{1}{r^2}\frac{\mathrm{d}\mathcal{Z}}{\mathrm{d}r}$$

where $\mathcal{Z}_L(r_I)$ = POTHAM(I, L) is a very smooth function. For orbital quantum numbers $l > L - 1$ the routine uses the last supplied POTHAM(I, L). Of course the option IPSEUD = -1 cannot account properly for exchange.

SS

is called by NAME. It reads and processes target orbital input supplied in numerical form in COLALG/IMPACT format — see [38,42]. Radial points r_I are stored into XR(I) of /SIMP/ and the statistical model electric charge $\mathcal{Z}_{\text{eff}}^{\text{SM}}(r_I)$ of Eq. (81) is temporarily saved in PX(I) of /POTVAL/ for use by routines TABORB and POTF. Numerical target orbital values $P_{nl}(r_I)$ and the associated $Q_{nl}(r)$ of Eq. (70) and Eq. (72) are read into UJ(I, K) and DUJ(I, K) of /ORBTLS/, while also setting up the address array IPOS($n, l+1$) = K. A blank or a KEY = -5 record terminates reading from unit INDATA.

Target orbitals are tabulated at points sufficient to support atomic structure calculations. This may not be good enough for collisional work, especially at higher energies, as the number of oscillations over a given radial stretch increases with the wave number k. Because the grid read by routine SS will be used as integration mesh for the continuum basis it is sometimes necessary to insert extra mesh points. This is controlled by the maximum collision energy MAXE, a compulsory input parameter. Starting with the outermost interval midpoints are inserted, and at these points $\mathcal{Z}_{\text{eff}}(r)$ is quadratically approximated and P and Q are *interpolated* as a 4-point Lagrange series; if necessary this procedure will be repeated until there are only two tabulation intervals left. 23 points over one period of the continuum function with the largest number ($\approx 2 *$ NRANG2) of nodes secures 16 points for a wave associated with the largest channel energy and thus at least 8 points in any integrand over one half-period.

It is not necessary to supply the wavefunctions far beyond a point r_{IC} where the effective charge has dropped to its residual value

$$\mathcal{Z}_{\text{eff}}(r \geq r_{\text{IC}}) = z \equiv Z - (N-1) \qquad \text{from} \quad \mathcal{Z}_{\text{eff}}(0) = Z$$

When supplying such truncated input one *must* specify RA as an input parameter. The program *extrapolates* P and Q between r_{b} and a mesh point nearest to |RA| as a Whittaker series:

$$P_{nl}(r) = \left(\frac{P_{nl}(r_{\text{b}})\ \mathrm{e}^{+\sqrt{-\epsilon}r_{\text{b}}}}{r_{\text{b}}^{\nu}\ f(\nu, l; r_{\text{b}})}\right)\left(\mathrm{e}^{-\sqrt{-\epsilon}r}\ r^{\nu}\ f(\nu, l; r)\right) \tag{87}$$

where

$$f(\nu, l; r) = \sum_{k=0}^{\text{KK}} \frac{a_k}{r^k}$$

is an asymptotically convergent series with coefficients

$$a_k = \left(\frac{l(l+1) - (\nu - k)(\nu - k + 1)}{2k\sqrt{-\epsilon}}\right) a_{k-1}$$

with $a_0 = 1$ and as usual

$$\nu \equiv n - \mu_l = \frac{z}{\sqrt{-\epsilon_{nl}}}$$

is the effective quantum number. All the quantities required can be extracted from P and Q at two tabulation points $r_b, r_{b-1} > r_{IC}$. The program chooses the last two tabulation points but one, which allows for a ready-made test: at the last point the supplied function value $P(r)$ is printed against its value computed from Eq. (87) for comparison — they must be virtually identical. Also printed are *computed* quantities such as KK, IC, $\mathrm{E} = \epsilon_{nl}$, and $\mathrm{A}(K) = a_K$ for $1 \le K \le$ KK; E may be checked against the eigenvalue ϵ_{nl} on the KEY = −7 header record, which precedes the P/Q records KEY = −6. The code exploits the identity

$$Q_{nl}(r) = \left(\epsilon_{nl} - 2 \cdot \mathrm{D}/r\right) P_{nl}(r) \qquad \text{if } r > r_{\mathrm{IC}}$$

and D is also printed. This value, normally an integer equal to the number of screening electrons, becomes a fractional value for 'PJS' type correlation functions satisfying a wave equation

$$\left(\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{l(l+1)}{r^2} + \frac{2\mathcal{Z}_{\mathrm{eff}}(r)}{r} + \epsilon_{nl}\right) P_{nl}(r) = 0$$

with

$$\mathcal{Z}_{\mathrm{eff}}^{\mathrm{PJS}}(r) = Z\left[\mathrm{e}^{-Zr/2} + \lambda_{nl}\left(1 - \mathrm{e}^{-Zr/2}\right)\right]$$

On the header record such orbitals are flagged as having 'scaling factors' $-(100n + \lambda_{nl})$.

We note that Eq. (87) supposes an exponential decay of $P_{nl}(r)$ controlled by a *single* eigenvalue ϵ_{nl}. As this excludes single-particle wavefunctions that have been Schmidt-orthogonalised in advance against other orbitals the user may supply a non-orthogonal set; SS will Schmidt-orthogonalise *after* extrapolating. Conversely a positive value in the position for ϵ_{nl} on the KEY = −7 header record would be interpreted as a flag not to Whittaker-extrapolate.

Three more points in conclusion:

1. To minimise numerical errors the input range is extended to the next full oscillation amplitude beyond r_{IC} unless, of course, the exponentially decaying tail has already started.
2. Specify a sufficiently large *negative* value RA when unsure, as SS will then automatically reduce RA to a value at which the most diffuse orbital has decayed to relative order TINORB = 0.001 — see DATA in SS.
3. For numerical reasons (restricted exponential range of real numbers) the extrapolation procedure simplifies once $|P_{nl}(r)| < 10^{-30}$ has been reached, which is typical for closed-shell 1s-electrons. Then SS extrapolates linearly to 0.0 at RA, with some safeguards for $Q_{nl}(r)$. Detailed printout on the right hand boundary of the three regions tells at a glance which of the two extrapolation ranges may eventually have collapsed to length zero.

In the special case of IPSEUD ≠ 0 the routine SS will accept more than one set KEY = −8 (preceded as usual by a KEY = −9 header record); the sets $\mathcal{Z}(r_i)$ are stored in successive orbital positions l of array POTHAM$(i, l+1)$.

STG1RD
reads and verifies all the input data described in Section 2.3, with the exception of alternative user-supplied radial orbital data, which are handled through a call to NAME (Section 2.4). For verification it prints the basic log file information and it sets secondary parameters. The choice of RA is made so that exchange between the scattering electron and the atomic target is negligible when the scattering electron coordinate $r >$ RA as in Eq. (11).

STO
called by STG1RD to check the orthonormality of the bound Slater-type basis orbitals, and aborts with an error message if not acceptable.

TABORB
creates a target orbital output file in SUPERSTRUCTURE COLALG/IMPACT format [38] — the inverse to the task of SS; the file is printed on unit number IPUNCH if NBUG5 = 2. The KEY = −9...KEY = −6 convention applies — see Section 2.4.2. One feature expected on the records of type KEY = −8 along with the radial grid may not be readily available: unless supplied in S.S. type input the effective Coulomb charge $\mathcal{Z}_{\text{eff}}(r)$ seen by a target electron is therefore *estimated* using the $(N-1)$-electron analogue of Eq. (80).

WRITAP
writes basic information onto the permanent output file ITAPE3 for processing by STG2 or passing through STG2 on to STGH.

2.2. Data files

The following is a summary of the data files required by STG1. The unit numbers and file names are defined within the program. Although the variables are part of the input data (for consistency with earlier versions of RMATRX) you need only supply dummy values i.e. set them to 0. The exception is ITAPE1 which should be set > 0 in the input data if you wish to read from 'STG1.POT'.

IREAD = 5 — 'STG1.INP'
File type: formatted sequential input.
Written or assembled by the user.
Read by routine STG1RD.
Description: N-electron and $(N+1)$-electron radial function data (see Section 2.3 for details)

IWRITE = 6 — 'STG1.OUT'
File type: formatted sequential output.
Written throughout STG1.
Read by user.
Description: line-printer or job output — the log file.

IPUNCH = 7 (only used if NBUG5 = 2) — 'ORBITAL.OUT'
File type: formatted sequential output.
Written by routine TABORB.
Read by routine SS or programs SUPERSTRUCTURE or COLALG.
Description: bound orbitals tabulated in S.S. format.

IDISC1 = 0 NOT USED
IDISC2 = 0 NOT USED
IDISC3 = 0 NOT USED

ITAPE1 = 1 (not normally used, i.e. if IPSEUD=0: input as 0) — 'STG1.POT'
File type: binary sequential input.
Written by model potential program.
Read by routine STG1RD.
Description: model potential
(see Section 2.5 for details).

ITAPE2 = 0 NOT USED

ITAPE3 = 3 — 'STG1.DAT'
File type: binary sequential output.
Written by routines WRITAP and GENINT.
Read by module STG2.
Description: basic information, multipole, one-electron and bound-bound two-electron integrals (see Section 2.7 for details).

ITAPE4 = 0 NOT USED
JDISC1 = 21 — 'RK.DAT'
File type: direct access output, record length 512 (8-byte) words.
Written by routine GENINT via DA2.
Read by module STG2 via DA2.
Description: bound-continuum and continuum-continuum two-electron integrals (see Section 2.8 for details).
JDISC2 = 0 NOT USED
INDATA = 10 (used only for S.S. input — 'ORBITAL.INP'
File type: formatted input.
Written by routine TABORB or programs SUPERSTRUCTURE or COLALG.
Read by routines SS.
Description: bound orbitals tabulated in S.S. format (see Section 2.4.2 for details).

2.3. *Input data on IREAD*

The user-supplied input data is read in routine STG1RD on input unit number IREAD. Free format is used, with one exception:

• FORMAT(18A4) for reading text into array TITLE.

If TITLE(1) ='S.S.' or 'CIV3' or 'STO-', then NAMELIST and alternative radial orbital input file options are invoked; see Section 2.4.

Summary of the data records (the variable names are described in the glossary in Section 9):

1. (TITLE(K),K=1,18)
   I/O units, described in Section 2.2:
2. IPUNCH, IDISC1, IDISC2, IDISC3, IDISC4,
   ITAPE1, ITAPE2, ITAPE3, ITAPE4, JDISC1
   Debug parameters, described in Section 2.6:
3. NBUG1, NBUG2, NBUG3, NBUG4, NBUG5, NBUG6, NBUG7, NBUG8, NBUG9
   Basic information:
4. ICOPY, ITOTAL, IPSEUD, (IRELOP(K),K=1,3), IZESP
5. If IZESP > 0:
   a. (ZESP(I),I=1,IZESP)
6. NELC, NZ, LRANG1, LRANG2, NRANG2, LAMAX, LAM, IBC, NPOT, LCB
7. (MAXNHF(L),L=1,LRANG1)
8. (MAXNLG(L),L=1,LRANG1)
9. Radial orbitals, as in Eq. (70):
   a. NCO
   b. (IRAD(J),J=1,NCO)
   c. (ZE(J),J=1,NCO)
   d. (C(J),J=1,NCO)
   repeat records a–d for L=1,LRANG1 and N=L,MAXNHF(L).
10. If IBC $\neq$ 0 then read boundary conditions:
    a. RA, BSTO
11. If |IBC| = 2 then read integration mesh, as in Eq. (77):
    a. NIX
    b. (IHX(I),I=1,NIX) (only if NIX > 0)
    c. (IRX(I),I=1,NIX) (only if NIX > 0)
    d. X, DELTA, ETA (HINT = X if NIX > 0)

12. If NPOT > 0 then read zero-order potential function (NPOT ≤ 6), as in Eq. (79):
    a. (IPOT(I), I=1,NPOT)
    b. (CPOT(I), I=1,NPOT)
    c. (XPOT(I), I=1,NPOT)
13. If IPSEUD > 0 and ITAPE1 > 0:
    a. L, (MAXNC(I),I=1,L)
14. If NPOT = 0 then the following card is read in routine POTF, provided NBUG6 = 1 or NELCOR > MAXELC = 18:
    a. (NSHELL(I), LSHELL(I), I=1,13)

Important note

Records 5 and 10–14 are optional; the inexperienced user is recommended not to invoke them.

2.4. *Option: alternative inputs NAMELIST and S.S.*

TITLE contains 72 characters of text and is printed out on unit IWRITE as a heading for the calculation. If the first four characters are:

S.S. then radial orbital input is read in the format described by Crees et al. [38] and generated by routine RADIAL of the SUPERSTRUCTURE code [43];

CIV3 then radial orbital data in CIV3 format [52] is expected;

STO- then radial orbital data in STG1 format is expected.

In each of these cases the NAMELIST format is invoked and this section replaces the descriptions given in Section 2.3. The NAMELIST option is described first, followed by the S.S. option.

Some basic parameters as described in Section 2.3 do not have to be explicitly input in the NAMELIST mode, as they can be derived from the radial orbital input. For example, NZ or Z is picked up as $\mathcal{Z}_{\text{eff}}(0)$ from the S.S orbital input, and NELC = N as $Z - \mathcal{Z}_{\text{eff}}(r_{\max}) + 1$.

Similarly, the orbital input also determines LRANG1 and the default values for RA and the integration mesh parameters NIX, IHX, IRX, HINT.

The variable names are described in the glossary in Section 9.

2.4.1. *NAMELIST option*

Summary of the data records:

1. (TITLE(K),K=1,18)
2. CIV3 input on unit INDATA. The CIV3 input is described in [52].
3. NAMELIST /STG1/ (the first three parameters are compulsory, the rest are optional):
   MAXLS,MAXPW,MAXE,
   IOUT,IOUTDA,INDATA,IPSEUD,
   NBUG1,NBUG2,NBUG3,NBUG4,NBUG5,NBUG6,NBUG7,NBUG8,NBUG9,
   KRELOP,LAM,LAMAX,MAXC,IBC,NPOT,LCB,ISMIT(L)
4. If IBC $\neq 0$ then read boundary conditions:
   a. RA, BSTO
5. If $|$IBC$| = 2$ then read integration mesh, as in Eq. (77):
   a. NIX
   b. (IHX(I),I=1,NIX) (only if NIX > 0)
   c. (IRX(I),I=1,NIX) (only if NIX > 0)
   d. X, DELTA, ETA (HINT = X if NIX > 0)
6. If NPOT > 0 then read zero-order potential function (NPOT ≤ 6), as in Eq. (79):
   a. (IPOT(I), I=1,NPOT)

b. (CPOT(I), I=1,NPOT)
c. (XPOT(I), I=1,NPOT)

7. S.S. input on unit INDATA. The S.S. option is described in the next section.
8. If IZESP > 0:
   a. (ZESP(I),I=1,IZESP)
9. If IPSEUD > 0 and ITAPE1 > 0:
   a. L, (MAXNC(I),I=1,L)
10. If NPOT = 0 then the following card is read in routine POTF, provided NBUG6 = 1 or NELCOR > MAXELC = 18:
    a. (NSHELL(I), LSHELL(I), I=1,13)

Important note
Records 4–6, 8–10 are optional; the inexperienced user is recommended not to invoke them.

2.4.2. Input of radial orbitals in S.S. format

The radial orbital input is read in the format described by Crees et al. [38] and generated by routine RADIAL of the SUPERSTRUCTURE code [43]. It can also be generated from Slater-type orbitals as an option by routine TABORB in STG1.

Summary of the S.S. data records on unit INDATA:
Tabulation header,
format(I5,I5,A13,I5,I4,I4,A24):
a. KEY=-9, NBOUND, LSTR, NPTS, NELC, NZ, TEXT;
Radial mesh and potential,
format(I5,2(I4,E14.7,E14.7),A11):
b. KEY=-8, I, XR(I), PX(I), I+1, XR(I+1), PX(I+1), TEXT;
repeat record b for I = 1, NPTS, 2;
Orbital header,
format(I5,I5,I5,I3,A62):
c. KEY=-7, K, NS, LS, TEXT;
Orbital tabulation,
format(I5,2(I4,E14.7,E14.7),A11):
d. KEY=-6, I, UJ(I,K), DUJ(I,K), I+1, UJ(I+1,K), DUJ(I+1,K), TEXT;
repeat record d for I = 1, NPTS, 2;
repeat records c–d for K = 1, NBOUND.

2.5. Input of model potential on ITAPE1

This occurs when IPSEUD > 0 and ITAPE1 > 0.
a. LRANG1,LRANG2, (MAXNC(I),I=1,LRANG1)
b. HINT,NIX, (IHX(I),I=1,NIX), (IRX(I),I=1,NIX)
c. LPOT, (LPOSX(I),I=1,LRANG2)
d. (POTHAM(J,I),J=1,NPTS)
where NPTS=IRX(NIX) and record d is repeated for I=1,LPOT.

2.6. Debug prints

Debugging prints are under the control of the NBUG parameters, specified in record 3 of the input data, or in the /NAMELIST/. Set these to zero in production runs, otherwise a large amount of output can be printed.
NBUG1 > 0 for debug printout from BASFUN. The intermediate integrations and energies are output.

NBUG2 = 0 NOT USED.
NBUG3 > 0 for debug printout from BASFUN. The arrays for the determination of the mismatch are output.
NBUG4 > 0 for tracing iterations in FINDER.
NBUG5
> 0 for a printout of overlap integrals, any Schmidt coefficients, and all orbitals at NBUG5 points of the integration mesh from BASORB;
= 1 for debugging bound orbital corrections in EVALUE and to write out RVAL from MESH (checks that the generated radial mesh gives the correct boundary radius RA);
> 1 for debugging the potential array POVALU from POTF;
= 2 for bound orbital printout to unit IPUNCH in S.S. format from TABORB.
NBUG6 is used in routine POTF to allow the reordering of the shells which determine the potential when NPOT = 0. The default ordering is 1s, 2s, 2p, 3s, 3p, 4s, 3d, 4p, 5s, 4d, 5p, 6s and 4f. If you wish to read in the arrays NSHELL and LSHELL which will change this ordering then you should set NBUG6 = 1.
NBUG7 = 1 for a dummy run to test that the input data does not exceed array sizes. The program jumps round all the calculations, so the run is very short, but goes through all the loops and prints out the required space and the current length of the arrays.
NBUG8
= 1 for printout of all bound-bound integrals from GENINT;
= 2 for printout as 1, plus bound-continuum integrals;
= 3 for printout as 2, plus continuum-continuum integrals.
NBUG9 = 1 for debug printout from SPNORB of the spin-orbit integrals.

For debugging purposes, LRANG2 may be set to 0, and computation is reduced to the requirements of target structure — a useful feature when running the Breit-Pauli code merely in order to obtain term-coupling coefficients (Saraph [69]) in module RECUPD. And within STG1 it may be chosen for fast preliminary runs to find an economical value of RA, as the program readily prints the relative magnitude of the most diffuse orbital.

2.7. Output on ITAPE3

The variable names are described in the glossary in Section 9.

Basic data from routine WRITAP:

1. NELC, NZ, LRANG1, LRANG2, NRANG2, LAMAX, ICODE, LAM, IZESP, (IRELOP(K),K=1,3);
2. (MAXPN(L),L=1,LRANG1), (MAXNLG(L),L=1,LRANG1), (MAXNC(L),L=1,LRANG1);
3. (EIGENS(N,L),N=1,NRANG2);
4. (ENDS(N,L),N=1,NRANG2+1);
   repeat records 3–4 for L=1,LRANG2;
5. RA, BSTO, H, DELTA, ETA, NIX;
6. ((COEFF(I,L),I=1,3),L=1,LRANG2).
   Multipole integrals Eq. (82) and Eq. (69) from routine GENINT:
7. IRK8, JRK8, IBBI;
8. (((IBBPOL(I,J,K),I=1,LRANG1),J=1,LRANG1),K=1,LAMIND),
   (((IBCPOL(I,J,K),I=1,LRANG1),J=1,LRANG2),K=1,LAMIND),
   (((ICCPOL(I,J,K),I=1,LRANG2),J=1,LRANG2),K=1,LAMIND),
   (RKSTO2(I),I=1,IRK8);
9. ((JBCPOL(I,J),I=1,LRANG1),J=1,LRANG2),
   ((JCCPOL(I,J),I=1,LRANG2),J=1,LRANG2),

(SKSTO2(J),J=1,JRK8), (BNORM(J),J=1,LRANG2).

One-electron integrals Eq. (78) and optionally Breit-Pauli integrals from routine GENINT:

10. IRK5;
11. (IST1(I),I=1,LRANG1), (ONEST1(I),I=1,IRK5);
    a. (RMASS1(I),I=1,IRK5) (only if IRELOP(1) > 0);
    b. (RSPOR1(I),I=1,IRK5) (only if IRELOP(3) > 0);
    c. IRK9 (only if IRELOP(2) > 0);
    d. (RDAR1(I),I=1,IRK9) (only if IRELOP(2) > 0);
12. IRK6;
13. (IST2(I),I=1,LRANG1), (ONEST2(I),I=1,IRK6);
    a. (RMASS2(I),I=1,IRK6) (only if IRELOP(1) > 0);
    b. (RSPOR2(I),I=1,IRK6) (only if IRELOP(3) > 0);
    c. IRK10 (only if IRELOP(2) > 0);
    d. (RDAR2(I),I=1,IRK10) (only if IRELOP(2) > 0);
14. IRK7;
15. (ONEST3(I,L),I=1,IRK7);
    a. (RMASS3(I,L),I=1,IRK7) (only if IRELOP(1) > 0);
    b. (RSPOR3(I,L),I=1,IRK7) (only if IRELOP(3) > 0 and L > 1);
    c. (RDAR3(I),I=1,IRK7) (only if IRELOP(2) > 0 and L = 1);

    repeat records 14–15 for $l+1$ =L=1,LRANG2.

    Two-electron integrals Eq. (73) from routine GENINT:
16. IRK1, IRK4;
17. (((ICTBB(I,J,K),I=1,LRANG1),J=1,LRANG1),K=1,I1),
    (ISTBB1(I),I=1,IRK4), (ISTBB2(I),I=1,IRK4), (RKSTO1(I),I=1,IRK1);
    (with I1=LRANG1*LRANG1);
18. IRK2, IRK3;
19. (((ICTBC(I,J,K),I=1,LRANG1),J=1,LRANG1),K=1,I1),
    (ISTBC1(I),I=1,IRK3), (ISTBC2(I),I=1,IRK3);
    (with I1=min(LRANG1*LRANG2, LRANG1*((LRANG1-1)*3+1));
20. JRK2, L, LP;
    repeat record 20 if JRK2 is negative:
21. (ICT(I), I=1,(I1+I2)*LRANG1*LRANG1);
    (with I1=min(2*LRANG1-1, L+LP+1); I2=min(LRANG1+L, LRANG1+LP))
    repeat records 20–21 for LP = L, LRANG2,
    repeat records 20–21 for L = 1, LRANG2.

2.8. *Output of two-electron radial integrals on JDISC1*

A direct access file 'RK.DAT' is used to store all the bound-continuum and continuum-continuum two-electron radial integrals generated in STG1, for use in module STG2. The direct access file is written and read by a call to routine DA2, which blocks a large array automatically into fixed record lengths, under the control of pointers. DA2 is called by routines GENINT in STG1, and RDINT in STG2.

The variable names are described in the glossary in Section 9.

Bound-continuum two-electron integrals Eq. (73):

1. (RKSTO2(I),I=1,IRK2).

Continuum-continuum two-electron integrals for each continuum L,LP combination:

```
  1  MNSTG2 STG2   AIJS   ----
  8                BOUND  ----
 33                DMEL   ----
 79                ISTG2  FACTT(*)
 80                       RECOV2
 81                       RME(*)
 82                       SHRIEK
 83                SETMX1 DA2
 84                       MATANS > 9
 85                       NJLJOD
 86                       PNTBG2 > 31
 87                       RDINT  DA2
 88                       SJ1QNT
 89                SETMXR MATANS > 9
 90                       NJLJOD
 91                       PNTBG2 > 31
 92                       RDINT  > 87
 93                       SJ1QNT
 94                SETUP  DMCON  > 36
 95                       RECOV2
 96                       SETCUP RECOV2
 97                STG2RD ----
105                WRITAP
```

Fig. 6. Calling tree for module STG2. The routines are given in alphabetical order within each branch **not** the order in which they are actually called. The routines marked with (*) are in module STGLIB.

2. (RKST02(I),I=1,abs(JRK2));
   (repeat record 2 by setting JRK2 negative if the number of integrals for a given L,LP combination exceeds the dimension of RKST02);
   repeat record 2 for LP = L, LRANG2;
   repeat record 2 for L = 1, LRANG2.

3. Module STG2

This module is the second stage in RMATRX1. It calculates *LS*-coupling matrix elements in the inner-region as in Eq. (14) and Eq. (48).

STG2 must be linked with STGLIB in order to form an executable program. Routines CG, CHOP, DRACAH, FACTT, FANO, HOWTS, HSLDR, INTACT, MEKEST, ORTHOG, REDUCE, RME, SETM, SETUPE, TENSOR and TRITST plus routines which they call are obtained from STGLIB.

Fig. 6–10 displays a flow diagram for the routines in STG2.

There are five main computational sections in STG2 (the controlling routines are named in brackets):
• initialisation and reading input files (STG2RD,ISTG2,WRITAP);
• solving the target-state (N-electron) problem (BOUND);
• setting up the $(N+1)$-electron Hamiltonian matrix (SETUP,SETMX1,SETMXR);
• evaluating the long-range potential coefficients (AIJS);
• setting up the $(N+1)$-electron dipole matrix (DMEL).

3.1. Routines

MNSTG2
is the program routine and contains all COMMON blocks used in STG2. It sets /MEMORY/ pointers, and

```
 1          AIJS   ALDAIJ DRACAH(*)
 2                        RME(*)
 3                        TRITST(*)
 4                 REDRAD FINMNT
 5                        RME(*)
 6                 SETUPE(*)
 7                 TENSOR(*)
```

Fig. 7. Calling tree for module STG2, AIJS section.

```
 8          BOUND  HSLDR(*)
 9                 MATANS MATRX  CHOP(*)
10                               DHO    FIN1BB
11                                      FIN1C
12                               HOWTS(*)
13                               ODHO   FIN1BB
14                                      FIN1BC
15                                      FIN1C
16                               ORTHOG(*)
17                               RKWTS  FANO(*)
18                                      MEKEST(*)
19                                      PRNTWT FINBB  INTECH
20                                             FINBC  INTECH
21                                             FINCC1 FINBB  > 19
22                                                    INTECH
23                                             FINCC2 FINCC1 > 21
24                                                    INTECH
25                                      REDUCE(*)
26                                      SETM(*)
27                                      USEEAV FANO(*)
28                                             INTACT(*)
29                                             PRNTWT > 19
30                 NJLJOD
31                 PNTBG2 VIJOUT
32                 SJ2QNT
```

Fig. 8. Calling tree for module STG2, BOUND section.

calls AASTG2.

AASTG2
is called by MNSTG2 and controls the STG2 computation, invoking the five main computational sections as summarised above.

There is a loop over the $(N+1)$-electron system $LS\pi$ symmetries to calculate Hamiltonian matrices and potential coefficients, which is completed before entering the dipole matrix routine DMEL.

AIJS
is the controlling routine for the evaluation and output to file ITAPE3 of the long-range potential coefficients coupling two channel eigenstates $\overline{\Phi}_i$ from Eq. (29):

$$\begin{aligned}\mathrm{CF}(i,j,\lambda) &= 2a_{ij}^{\lambda} \\ &= 2\langle \overline{\Phi}_i(x_1\ldots x_N, \hat{r}_{N+1}\sigma_{N+1}) \mid \sum_{n=1}^{N} r_n^{\lambda} P_{\lambda}(\cos\theta_{n,N+1}) \mid \overline{\Phi}_j(x_1\ldots x_N, \hat{r}_{N+1}\sigma_{N+1})\rangle\end{aligned} \tag{88}$$

```
33            DMEL   CG(*)
34                   CHEKTP DA2
35                          RECOV2
36                   DMCON  DA2
37                   DMELBB FINMNT
38                          RME(*)
39                          SETUPE(*)
40                          TENSOR(*)
41                   DMELBC RME(*)
42                          SETINI
43                          SETUPE(*)
44                          TENSOR(*)
45                   DMELBD RME(*)
46                          SETINI
47                          SETUPE(*)
48                          TENSOR(*)
49                   DMELCB RME(*)
50                          SETFIN
51                          SETUPE(*)
52                          TENSOR(*)
53                   DMELCC FINMNT
54                          RME(*)
55                          SETFIN
56                          SETINI
57                          SETUPE(*)
58                          TENSOR(*)
59                   DMELCD RME(*)
60                          SETFIN
61                          SETINI
62                          SETUPE(*)
63                          TENSOR(*)
64                   DMELDB RME(*)
65                          SETFIN
66                          SETUPE(*)
67                          TENSOR(*)
68                   DMELDC RME(*)
69                          SETFIN
70                          SETINI
71                          SETUPE(*)
72                          TENSOR(*)
73                   DMELDD FINMNT
74                          RME(*)
75                          SETFIN
76                          SETINI
77                          SETUPE(*)
78                          TENSOR(*)
```

Fig. 9. Calling tree for module STG2, DMEL section.

for $i, j = 1,$NCHAN and $\lambda = 1,$LAMAX (the highest multipole order LAMAX is input from module STG1), where $\cos\theta_{n,N+1} = \hat{\boldsymbol{r}}_n \cdot \hat{\boldsymbol{r}}_{N+1}$.

The expression is evaluated by first writing it in tensor notation as

$$a_{ij}^{\lambda} = \langle \alpha_i L_i S_i l_i \tfrac{1}{2}; LM_L SM_S \mid \boldsymbol{M}_\lambda . \boldsymbol{C}_\lambda \mid \alpha_j L_j S_j l_j \tfrac{1}{2}; LM_L SM_S \rangle$$

where

```
 97          STG2RD CHEKTP > 34
 98                 CONFIG CONPED CONQN  CONSH  CONSTO CONTST
 99                                                    RECOV2
100                                             CONTST
101                        CONSTO > 98
102                        RECOV2
103                 COPYTP RECOV2
104                 RECOV2
```

Fig. 10. Calling tree for module STG2, STG2RD section.

$$M^{\mu}_{\lambda}=\left(\frac{4\pi}{2\lambda+1}\right)^{1/2}\sum_{n=1}^{N} r^{\lambda}_{n}Y^{\mu}_{\lambda}(\hat{r}_n)\quad\text{and}\quad C^{\mu}_{\lambda}=\left(\frac{4\pi}{2\lambda+1}\right)^{1/2}Y^{\mu}_{\lambda}(\hat{r}_{N+1})$$

Using Eq. (15.6) of Fano and Racah [45] this can be reduced to

$$a^{\lambda}_{ij}=(-1)^{L+L_i+1/2(\lambda-l_i-l_j)}\,(2l_i+1)^{1/2}\,C(l_i\lambda l_j;00)\,W(L_iL_jl_il_j;\lambda L)\times\mathtt{CFADD}\tag{89}$$

Note that CFADD does not depend directly on the l_i and l_j channel angular momenta, but on the target states themselves (L_i and L_j). Since the N-electron states are in general multi-configurational, CFADD is calculated as a weighted sum of contributions from each configuration in the expansion of Eq. (7):

$$\mathtt{CFADD}=\sum_{k=1}^{\mathtt{NTCON}(i)} b_{ik}\sum_{k'=1}^{\mathtt{NTCON}(j)} b_{jk'}\,(\alpha_kL_i\;||\;M_\lambda\;||\;\alpha_{k'}L_j)$$

where in this and in later equations the reduced matrix elements are defined by

$$\langle\alpha_iL_iM_{L_i}\mid M^{\mu}_{\lambda}\mid\alpha_jL_jM_{L_j}\rangle=\frac{C(L_j\lambda L_i;M_{L_j}\mu)}{(2L_i+1)^{1/2}}(\alpha_iL_i\;||\;M_\lambda\;||\;\alpha_jL_j)$$

The summations over the atomic configurations involved in the reduced matrix element are carried out in this routine. REDRAD is called to retrieve the multipole radial integrals to evaluate the reduced matrix element, and SETUPE and TENSOR are called to evaluate the angular and spin factors. Thus CFADD is passed to ALDAIJ to complete the evaluation of Eq. (89).

ALDAIJ
is called from AIJS for a given pair of target states. It takes as input the reduced matrix element CFADD. It loops over channels l_i and l_j coupled to the two target states (since CFADD is independent of l_i and l_j). It then calls function RME for the Clebsch-Gordan (C) and DRACAH for the Racah (W) coefficients in Eq. (89) (these coefficients are independent of the target configuration sum) to form CF$(i,j,\lambda)=2a^{\lambda}_{ij}$, which is returned to AIJS.

BOUND
is the controlling routine for solving the target-state problem in LS-coupling by calculating the N-electron Hamiltonian (H^N) matrix elements involving the target basis in Eq. (7):

$$H_{kk'}=\langle\phi_k(x_1\ldots x_N)\mid H^N\mid\phi_{k'}(x_1\ldots x_N)\rangle$$

which is evaluated by calling MATANS using the radial integrals transferred from module STG1. There are then three modes of operation, depending on the value of JRELOP(3) read from the user input to STG2.

JRELOP(3) = 0. In LS-coupling runs the configuration mixing coefficients b_{ij} in Eq. (7) and energy levels E^N_i are then obtained by diagonalizing the N-electron Hamiltonian matrix as in Eq. (6) by calling HSLDR:

$$\langle \Phi_i \mid H^N \mid \Phi_{i'} \rangle = \delta_{ii'} E_i^N$$

The routine will automatically locate all states of the same $SL\pi$ symmetry and assign a different eigenvector in turn to each of these states, starting with the lowest eigenvalue. The target states do not need to be input in any particular order in STG2, and the original ordering is preserved here; module STGH will eventually reorder the states in ascending energy order.

JRELOP(3) = 1. In Breit-Pauli runs these matrix elements are written to the STG2 output file ITAPE3 for module RECUPD, and routine BOUND is terminated at this point.

JRELOP(3)= −1. This is a special option for term-coupling coefficients only, and should not be used for production runs. Both the above options are invoked, so that the output to ITAPE3 contains both the eigenvalues and eigenvectors in *LS*-coupling and the matrix elements themselves, for diagonalizing in the intermediate-coupling scheme in module RECUPD. This will enable term-coupling coefficients to be obtained from RECUPD.

CHEKTP
reads basic data together with the multipole, one- and two-electron radial integrals produced in module STG1 from files ITAPE1 and JDISC1. Most of the data, including all bound-bound integrals, are stored in common block arrays for use in STG2. However, the number of integrals involving continuum orbitals can be very large, so CHEKTP uses the following strategy for their storage.

Multipole integrals. Although the bound-bound multipole integrals are stored in /INSTO5/ (they are required in routine AIJS), the remaining multipole integrals on ITAPE1 are not needed until the execution of routine DMEL, so CHEKTP is called again from there to re-read these integrals.

Bound-continuum and continuum-continuum two-electron integrals. These are stored on the JDISC1 file (RK.DAT) from module STG1. CHEKTP would like to store the entire contents of the file in the first locations of /MEMORY/, if there is enough space (&MEM). Any integrals not so stored have to be accessed from the JDISC1 file during execution of STG2, with possible degradation of performance due to the I/O overhead. It should not affect the results though. Pointers ITAPBC and ITAPST are set to enable integrals to be retrieved from /MEMORY/ (pointers positive) or file (pointers negative).

CONFIG
is the controlling routine for generation of configurations and associated coupling scheme data for states with given total angular momentum, spin and parity. CONFIG is called by STG2RD both for the target *N*-electron configurations ϕ_k in Eq. (7) and for the $(N+1)$-electron configurations χ_j in Eq. (12). The user can constrain the choice of the configurations by specifying the minimum and maximum number of electrons required in each shell, and the number of electron excitations allowed from given basic configurations; this is specified in the STG2 input data read in STG2RD. CONFIG loops over all possible distributions of electrons within the available shells, calls CONPED and returns coupling schemes for the valid configurations.

Because of a restriction on the coupling scheme arrays (imposed because of a limitation on fractional parentage coefficients in STG2), the CONFIG package does not produce configurations with more than 2 electrons in shells with orbital angular momenta $l \geq 3$.

CONPED
is part of the CONFIG package. It is called by CONFIG for a given electron distribution, tests whether this is consistent with the constraints imposed by the user, calls CONQN and returns coupling schemes for the valid configurations.

CONQN
is part of the CONFIG package. It is called by CONPED for a given electron distribution, loops over allowed quantum numbers for each shell, calls CONSH and returns coupling schemes.

CONSH
is part of the CONFIG package. It is called by CONQN for a given set of shell quantum numbers, determines the coupling between the shells, calls CONSTO and returns coupling schemes.

CONSTO
is part of the CONFIG package. It is called by CONSH, or can be called directly by CONFIG, for a given coupling scheme, which is stored in the appropriate STG2 arrays on return. Further options in CONSTO include: the printing out of configuration data under the control of a debug parameter (IBUG7); the output of data to file (IPUNCH); the reading of data from a prepared input file (JREAD); the deletion of specified configurations (ICUT). More detail on these options can be found in the description of the relevant STG2 input data parameters (in brackets above) in Section 3.3.

CONTST
is part of the CONFIG package. It is called by CONSTO for a given coupling scheme only if configuration data has already been pre-prepared, and returns OK=.TRUE. only if the configuration has the correct total angular momentum and spin.

COPYTP
positions the binary input file (ITAPE2), containing Hamiltonian matrices and asymptotic coefficients written to file (ITAPE3) by a previous run of STG2, in preparation for a restart.

DA2
is used to read input file 'RK.DAT' (JDISC1) created in module STG1. It is also used to write/read the scratch file IDISC1. See the description of this routine in Section 2.

DHO
is called by MATRX to calculate the one-electron matrix element when the configurations φ_j are identical; i.e. for the diagonal elements of the Hamiltonian matrix, returned in HOMAT in common block /CONMX/. The integral is evaluated as explained in the description of MATRX by multiplying the angular and radial parts and summing over the interacting shells:

$$\langle \varphi_j \mid H_0 \mid \varphi_j \rangle = \sum_\sigma x(\sigma,\sigma) \left\langle U_{n_\sigma l_\sigma} \left| -\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{Z}{r} + \frac{l_\sigma(l_\sigma+1)}{2r^2} \right| U_{n_\sigma l_\sigma} \right\rangle$$

where H_0 is the one-electron operator, $x(\sigma,\sigma)$ is just the number of electrons and $U_{n_\sigma l_\sigma}$ the radial orbital for the shell labelled σ. The radial one-electron integral is found by calling the FIN1... routines, with the mass-correction and Darwin terms included if required.

If an N-electron target configuration ϕ_k from Eq. (7) is being considered, the result is returned as a single element in HOMAT(1,1).

If the $(N+1)$-electron basis of Eq. (12) is involved, the matrix elements are returned in ((HOMAT(j,j'), j = 1, ILIMIT), j' = 1, JLIMIT), where:

ILIMIT=JLIMIT=1 for bound-bound;
ILIMIT=JLIMIT=NRANG2 for continuum-continuum.

In the latter case, only the diagonal elements are defined in DHO.

DMCON
is called by SETUP to store (when KEY = 2), and by DMEL to retrieve (when KEY = 1), $(N+1)$-electron configuration and channel data for each $SL\pi$ symmetry, for calculation of dipole matrices (i.e. only used if IPOLPH = 2). Storage can be either on disk (IDISC1) via a call to DA2, or in /MEMORY/ after the radial

integrals, if there is sufficient space (&MEM).

DMEL
is the controlling routine for the evaluation of the reduced dipole length and velocity matrix elements Eq. (48) for all dipole allowed transitions between the specified $SL\pi$ symmetries. DMEL is only activated if IPOLPH ≥ 2 as specified in the STG2 input data. The dipole matrix elements are stored in arrays DEL and DEV (length and velocity), and are written out to the STG2 binary output file ITAPE4 as specified in Section 3.6, for use in bound-bound, bound-free, free-free and polarizability calculations. Additional information is also calculated in DMEL and written to ITAPE4: the Buttle correction to the dipole matrix; matrices required for the outer-region contribution; and certain Clebsch-Gordan coefficients.

The total wavefunction in the inner region on both sides of the matrix is expanded in the form of Eq. (12). Let φ_λ denote collectively the antisymmetrised basis functions in Eq. (12). The inner region dipole matrix elements take the form of Eq. (48):

$$D_{\lambda\lambda'} = (\varphi_\lambda(x_1 \dots x_{N+1}) \,||\, \boldsymbol{D} \,||\, \varphi_{\lambda'}(x_1 \dots x_{N+1}))$$

Both length and velocity forms of the dipole operator $\boldsymbol{D}$ are considered:

$$D_{\rm L} = \sum_{n=1}^{N+1} z_n \quad \text{and} \quad D_{\rm V} = -\sum_{n=1}^{N+1} \frac{\partial}{\partial z_n}$$

Routine CHEKTP is first re-called to read the radial multipole integrals from file ITAPE1. Then the outer loop over dipole allowed transitions is entered, and DMCON is called to retrieve $(N+1)$-electron configuration and channel data for the two states involved.

Routines DMELCC, DMELCB, DMELBC and DMELBB are then called in sequence to calculate continuum-continuum (CC), continuum-bound (CB), bound-continuum (BC) and bound-bound (BB) dipole matrix elements respectively. When continuum basis terms are involved, the routine call is inside the channel loop (i in Eq. (12)). The arrays DEL and DEV are overwritten in each call to these routines by the appropriate block of matrix elements, which are written to the STG2 binary output file ITAPE4 as specified in Section 3.6. The dipole matrix elements are thus evaluated in the following blocks:

$$\text{DEL}(j,j') = (\varphi_j \,||\, D_{\rm L} \,||\, \varphi_{j'}) \quad \text{and} \quad \text{DEV}(j,j') = (\varphi_j \,||\, D_{\rm V} \,||\, \varphi_{j'}) \tag{90}$$

In the routines called by DMEL, FINMNT is called to retrieve the dipole radial integral and SETUPE and TENSOR to evaluate the angular and spin factors in the reduced matrix element which is expressed in the form

$$(\alpha LS \,||\, \sum_{n=1}^{N+1} T(n) \,||\, \alpha' L' S') = \text{VSHELL}(1) * (l_\rho \,||\, T \,||\, l_\sigma)$$

where $T(n)$ is a tensor operator which operates on the nth electron, ρ and σ denote the interacting shells. Routine TENSOR calculates VSHELL(1) and the remaining angular contribution is evaluated in function RME.

DMEL also calculates the Buttle correction to the dipole matrix. These corrections for a one channel problem have been discussed by Yu Yan and Seaton [85]. Generalising the theory to the many-channel case, if the Buttle corrections to the radial functions are ΔU_i, then a Buttle correction to the total wavefunction can be expanded in the form given by Eq. (6.23) of Berrington et al. [11]:

$$\Delta\psi_i = \mathcal{A}\,\overline{\Phi}_i\,\Delta U_i \tag{91}$$

Routines DMELCD, DMELBD, DMELDC, DMELDB and DMELDD are called to calculate the continuum-Buttle, bound-Buttle, Buttle-continuum, Buttle-bound and Buttle-Buttle dipole matrix elements respectively, and these data are written to the STG2 output file ITAPE4 as described in Section 3.6.

DMEL also writes to ITAPE4 the arrays AC, BLC and BVC calculated in routine DMELCC; these matrices are defined by Eq. (53) and are required for determining the outer region contribution to the dipole matrix. Note a confusion in notation: $B_{ii'}$ and $A_{ii'}$ in Eqs. (9.7) and (9.8) of Berrington et al. [11] correspond to AC(i,i') and BLC(i,i') respectively in the program;

$$\begin{aligned} \mathrm{AC}(i,i') &= (\overline{\Phi}_i \,||\, \hat{r} \,||\, \overline{\Phi}_{i'}) \\ BLC(i,i') &= (\overline{\Phi}_i \,||\, \mathbf{R} \,||\, \overline{\Phi}_{i'}) \qquad (92) \\ BVC(i,i') &= (\overline{\Phi}_i \,||\, \tfrac{\mathrm{d}}{\mathrm{d}r} \,||\, \overline{\Phi}_{i'}) \qquad (93) \end{aligned}$$

where i and i' are the channel indices in Eq. (52).

Finally, DMEL calls routine CG to form the following Clebsch-Gordan coefficients:

$$\mathrm{CGC}(m) = (2L+1)^{1/2}\, C(L'LL;m0) \qquad \text{where } m = 1, \min(L', L) \qquad (94)$$

which are also written to ITAPE4. In these coefficients, which are required in converting the reduced matrix elements of Eq. (47) to the matrix element defined by Eq. (44) for polarizability calculations, L' and L are the total orbital angular momentum LRGL of the initial and final states.

DMELBB
is part of the DMEL package, and calculates the bound-bound dipole matrix elements given by Eq. (90), with $j = 1,$NCFGP and $j' = 1,$NCFGP$'$. Matrix DEL returns the length form; similarly DEV returns the velocity form.

DMELBC
is part of the DMEL package, and calculates the bound-continuum dipole matrix elements given by Eq. (90), with $j = 1,$NCFGP and $j' = 1,$NRANG2 for a given continuum channel i'. Since the N-electron states are in general multiconfigurational, the matrix elements are evaluated as a weighted sum over configurations in the expansion of Eq. (7); matrix DEL returns the length form:

$$\mathrm{DEL}(j,j') = \sum_{k'=1}^{\mathrm{NTCON}(i')} b_{i'k'} (\chi_j \,||\, D_{\mathrm{L}} \,||\, \mathcal{A}\overline{\phi}_{k'} \tfrac{1}{r} u_{i'j'}) \qquad (95)$$

similarly DEV returns the velocity form. The function $\overline{\phi}_k$ is formed by coupling a target configuration ϕ_k from Eq. (7) with the angular and spin functions of the continuum electron to give the total angular momentum, spin and parity of the initial state. Routine SETINI is called for each configuration to extend the coupling scheme to include the continuum electron.

DMELBD
is part of the DMEL package and calculates the bound-Buttle dipole matrix elements between the bound basis terms of Eq. (12) and the Buttle correction basis terms of Eq. (91).

DMELCB
is part of the DMEL package, and calculates the continuum-bound dipole matrix elements given by Eq. (90), with $j = 1,$NRANG2 and $j' = 1,$NCFGP$'$ for a given continuum channel i. Matrix DEL returns the length form; similarly DEV returns the velocity form. This routine is similar to DMELBC and proceeds as in Eq. (95), except that the final and initial states are reversed. As the continuum basis term now belongs to the final state, routine SETFIN is called for each N-electron configuration to extend the coupling scheme to include the continuum electron.

DMELCC
is part of the DMEL package, and calculates the continuum-continuum dipole matrix elements given by Eq.

(90), with $j = 1,$NRANG2 and $j' = 1,$NRANG2 between continuum channels i and i'. Since the N-electron states are in general multiconfigurational, the matrix elements are evaluated as weighted sums over configurations in the expansion of Eq. (7); matrix DEL returns the length form:

$$\mathrm{DEL}(j,j') = \sum_{k=1}^{\mathrm{NTCON}(i)} b_{ik} \sum_{k'=1}^{\mathrm{NTCON}(i')} b_{i'k'} (\mathcal{A}\overline{\phi}_k \tfrac{1}{r} u_{ij} \;||\; D_{\mathrm{L}} \;||\; \mathcal{A}\overline{\phi}_{k'} \tfrac{1}{r} u_{i'j'})$$

similarly DEV returns the velocity form. The functions $\overline{\phi}_k$ are formed by coupling a target configuration ϕ_k from Eq. (7) with the angular and spin functions of the continuum electron to give the total angular momentum, spin and parity of the final and initial states involved. Routines SETFIN and SETINI are called for each configuration to extend the coupling scheme to include the continuum electron, for the final and initial state respectively.

The dipole matrix between two sets of continuum basis terms where the continuum electrons have equal angular momenta is in general a diagonal matrix. Moreover, the dipole matrix elements on the diagonal are non-zero only if the N-electron configurations differ by no more than one electron and if the channel angular momentum $l_i = l_{i'} \pm 1$. The angular contribution to such a non-zero diagonal element is evaluated in the usual way, but the radial integral corresponds to the electron transition between the bound levels and is found by calling FINMNT.

DMELCC also calculates and returns

$$\mathrm{ACOEF} = \mathrm{AC}(i,i') \quad \mathrm{BLCOEF} = \mathrm{BLC}(i,i') \quad \mathrm{BVCOEF} = \mathrm{BVC}(i,i')$$

between the given continuum channels i and i' as defined in Eqs.(92–93) and discussed in Section 9.3 of Berrington et al. [11]. The coefficient ACOEF, which is purely algebraic, is non-zero only if the channels i and i' belong to the same target state, and if the channel angular momentum $l_i = l_{i'} \pm 1$. The BLCOEF and BVCOEF coefficients are non-zero only if there is a dipole allowed transition between the N-electron target states belonging to channels i and i' and if $l_i = l_{i'}$.

DMELCD
is part of the DMEL package and calculates the continuum-Buttle dipole matrix elements between the continuum basis terms of Eq. (12) and the Buttle correction basis terms of Eq. (91), for a given continuum channel i.

DMELDB
is part of the DMEL package and calculates the Buttle-bound dipole matrix elements between Buttle correction basis terms of Eq. (91) and the bound basis terms of Eq. (12).

DMELDC
is part of the DMEL package and calculates the Buttle-continuum dipole matrix elements between the Buttle correction basis terms of Eq. (91) and the continuum basis terms of Eq. (12), for a given continuum channel i.

DMELDD
is part of the DMEL package and calculates the Buttle-Buttle dipole matrix elements between the Buttle correction basis terms of Eq. (91).

FIN1BB,FIN1BC,FIN1C
are called from routines DHO and ODHO to find a bound-bound(BB), a bound-continuum(BC) or an array of continuum-continuum(C) one-electron radial integrals. These are stored in the ONEST1, ONEST2 and ONEST3 arrays respectively, which are read by routine CHEKTP from the binary input file ITAPE1 (produced by module STG1). The routines also find and add in the mass-correction and Darwin terms if required.

FINBB
is called from routine PRNTWT to find a bound-bound two-electron radial integral in the RKST01 array, which was read by routine CHEKTP from the binary input file ITAPE1 (produced by module STG1). The (n,l) values of the four orbitals involved are input in common block /NJLJ/. Returns RKMAT(1,1) in /RKMATX/.

FINBC
is called from routine PRNTWT to find bound-continuum two-electron radial integrals in the RKST02 array, which was read by routine RDINT from the direct access input file JDISC1 (produced by module STG1). The (n,l) values of the four orbitals involved are input in common block /NJLJ/; the fourth orbital is continuum. Returns (RKMAT(1,j'),j' = 1,JLIMIT) in /RKMATX/.

FINCC1
is called from routine PRNTWT to find diagonal $(j = j')$ continuum-continuum two-electron radial integrals in the RKST02 array, which was read by routine RDINT from the direct access input file JDISC1 (produced by module STG1). The continuum angular momenta l_i and $l_{i'}$ may be different; when l_i=$l_{i'}$ the case may just involve bound orbitals and FINBB is called. The (n,l) values of the four orbitals involved are input in common block /NJLJ/; the second and fourth orbitals are continuum. Returns (RKMAT(j,j),j = 1,ILIMIT) in /RKMATX/.

FINCC2
is called from routine PRNTWT to find off-diagonal $(j \neq j')$ continuum-continuum two-electron radial integrals in the RKST02 array, which was read by routine RDINT from the direct access input file JDISC1 (produced by module STG1). The (n,l) values of the four orbitals involved are input in common block /NJLJ/; the second and fourth orbitals are continuum. Returns ((RKMAT(j,j'),j = 1,ILIMIT),j' = 1,JLIMIT) in /RKMATX/.

FINMNT
is called from routines DMELBB, DMELCC, DMELDD and REDRAD to find a multipole radial integral from the RKST02 array. The (n,l) values of the two orbitals concerned and the multipole order K are input as arguments: if $K = 1$ then the dipole length and dipole velocity integrals are returned; if $K > 1$ only the Kth pole length integral is returned.

INTECH
interchanges the two sets of quantum numbers (n_1,l_1) and (n_2,l_2). It is required in finding the appropriate continuum-continuum two-electron radial integral from the RKST02 array in the FIN... routines.

ISTG2
is called once from AASTG2 for initialisation:
- calls routines to store factorials in /FACT/ and /FACTT/;
- stores Kronecker delta function in /KRON/;
- sets up a pointer array in /SYMTX/ to store a symmetric matrix as a single array;
- calls RME to store coefficients $(l \,||\, C_k \,||\, l')$ for all l, l' and k in /CSTORE/.

MATANS
sets up the call to calculate a Hamiltonian matrix element, which is returned in array AME in common block /ELEMS/. Let φ_j be an electronic configuration from either Eq. (7) or Eq. (12), and let H be the appropriate Hamiltonian operator. Then

$$\text{AME}(j,j') = \langle \varphi_j \mid H \mid \varphi_{j'} \rangle \tag{96}$$

If MATANS is called from routine BOUND for the N-electron Hamiltonian, φ_j represents a single configuration ϕ_k from Eq. (7), and the matrix element is returned in AME(1,1). If it is called from the three sections of routine SETMX1 and SETMXR for the $(N+1)$-electron Hamiltonian, φ_j can be a single bound basis term or the continuum basis from Eq. (12); the matrix elements are returned in ((AME(j,j'), j = 1, ILIMIT), j' = 1, JLIMIT), where:

ILIMIT=JLIMIT=1 for bound-bound;
ILIMIT=1, JLIMIT=NRANG2 for bound-continuum;
ILIMIT=JLIMIT=NRANG2 for continuum-continuum.

In the case of continuum-continuum matrix elements when $l_i = l_{i'}$, it separates out the evaluation of the off-diagonal ($j \neq j'$) and diagonal ($j = j'$) elements. In the latter case the calculation is simplified because the two continuum orbitals are identical; the number of occupied shells (IHSH) is reduced by one and the coupling scheme contracted.

MATRX
is called by MATANS to evaluate a Hamiltonian matrix element, which is returned in array AME in common block /ELEMS/. Let H_0 be the one-electron operator and V the two-electron operator in the Hamiltonian in Eq. (96). AME is therefore evaluated as the sum of the two contributions:

$$\mathrm{AME}(j,j') = \langle \varphi_j \mid H_0 + V \mid \varphi_{j'} \rangle = \mathrm{HOMAT}(j,j') + \mathrm{VMAT}(j,j') \tag{97}$$

Moreover, the angular and radial integrals are separable:

$$\mathrm{HOMAT}(j,j') = \langle \varphi_j \mid H_0 \mid \varphi_{j'} \rangle = \sum_{\sigma\sigma'} x(\sigma,\sigma')\; Q(n_\sigma l_\sigma, n_{\sigma'} l_{\sigma'})\; \delta_{l_\sigma l_{\sigma'}} \tag{98}$$

where x is the angular and spin integral calculated in routine HOWTS which is multiplied by the one-electron radial integral Q in routines ODH0 and DH0;

$$\mathrm{VMAT}(j,j') = \langle \varphi_j \mid V \mid \varphi_{j'} \rangle = \sum_{\rho\sigma\rho'\sigma' k} y(\rho,\sigma,\rho',\sigma',k)\; R^k(n_\rho l_\rho, n_\sigma l_\sigma, n_{\rho'} l_{\rho'}, n_{\sigma'} l_{\sigma'}) \tag{99}$$

where y is the angular and spin integral calculated in routine FANO which is multiplied by the two-electron radial integral R^k in routine PRNTWT, invoked by calling RKWTS.

The ρ and σ indices in Eq. (98) and Eq. (99) label the interacting shells. ORTHOG is called to test for simple orthogonality of the configurations which would lead to zero matrix elements, and CHOP is called to remove shells whose interaction arises purely as an average energy.

NJLJOD
is called by routines BOUND, SETMX1 and SETMXR to put the NJ and LJ arrays, which hold the n and l values of the two configurations on each side of a matrix element, into a standard order.

ODH0
is called by MATRX and calculates the one-electron matrix element when the configurations φ_j and $\varphi_{j'}$ differ by one orbital; i.e. for the one-electron contribution to the off-diagonal elements of the Hamiltonian matrix, returned in HOMAT in common block /CONMX/. The integral is evaluated as explained in the description of routine MATRX by multiplying the angular and radial parts and summing over the interacting shells:

$$\langle \varphi_j \mid H_0 \mid \varphi_{j'} \rangle = \sum_{\sigma\sigma'} x(\sigma,\sigma') \left\langle U_{n_\sigma l_\sigma} \left| -\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{Z}{r} + \frac{l_\sigma(l_\sigma+1)}{2r^2} \right| U_{n_{\sigma'} l_{\sigma'}} \right\rangle \delta_{l_\sigma l_{\sigma'}}$$

where H_0 is the one-electron operator, $x(\sigma,\sigma')$ the angular integral involving shells labelled σ and σ' which is calculated in routine RKWTS and input into ODH0, and $U_{n_\sigma l_\sigma}$ the radial orbital for the shell labelled σ.

The radial one-electron integral is found by calling the FIN1... routines, with the mass-correction and Darwin terms included if required.

If an N-electron target configuration ϕ_k from Eq. (7) is being considered, the result is returned as a single element in HOMAT(1,1).

If the $(N+1)$-electron basis of Eq. (12) is involved, the matrix elements are returned in ((HOMAT(j,j'), j = 1,ILIMIT), j' = 1,JLIMIT), where:

ILIMIT=JLIMIT=1 for bound-bound;
ILIMIT=1, JLIMIT=NRANG2 for bound-continuum;
ILIMIT=JLIMIT=NRANG2 for continuum-continuum.

PNTBG2
is called from BOUND, SETMX1 and SETMXR to print out the angular momentum coupling before calculating the Hamiltonian matrix; only invoked if IBUG9 ≥ 4, i.e. for debug purposes.

PRNTWT
is called from the RKWTS package to form the two-electron contribution to the Hamiltonian matrix element as in Eq. (99) which is returned in array VMAT in common block /CONMX/. The two-electron radial integrals in array RKMATX are located by calling the FIN... routines, and are multiplied through by the appropriate angular and spin integrals, input in the arrays AMULT and BMULT in common block /XATION/ for the direct and exchange integrals respectively.

RDINT
is called from SETMX1 and SETMXR to read the bound-continuum or continuum-continuum two-electron radial integrals for given continuum angular momenta $(l_i, l_{i'})$, from /MEMORY/ or from the direct access input file (JDISC1) via a call to DA2, depending on the value of the pointers ITAPBC, ITAPST defined in routine CHEKTP.

RECOV2
See the description of this routine in Section 2.

REDRAD
is called from AIJS to calculate the product of the reduced matrix element (obtained from calling function RME) and the radial multipole integral (retrieved by calling FINMNT) between two orbitals in the evaluation of Eq. (89).

RKWTS
is described by Hibbert [50,51]. It calculates the angular and spin weighting factors for the two-electron contribution to the Hamiltonian matrix element between configurations φ_j and $\varphi_{j'}$ using the method of Fano [44]. It is called by MATRX and calls routine PRNTWT to return VMAT in common block /CONMX/. The integral is evaluated as explained in the description of MATRX (see Eq. (99)) by multiplying the angular and radial parts and summing over the interacting shells:

$$\langle \varphi_j \mid V \mid \varphi_{j'} \rangle = \sum_{\rho\sigma\rho'\sigma' k} y(\rho,\sigma,\rho',\sigma',k) \left\langle U_{n_\rho l_\rho} U_{n_\sigma l_\sigma} \left| \frac{r_<^k}{r_>^{k+1}} \right| U_{n_{\rho'} l_{\rho'}} U_{n_{\sigma'} l_{\sigma'}} \right\rangle$$

where V is the two-electron operator, $y(\rho,\sigma,\rho',\sigma')$ the angular and spin integral involving shells labelled ρ, σ ρ' and σ' which is calculated in terms of Racah algebra by routine FANO or USEEAV, $U_{n_\sigma l_\sigma}$ the radial orbital for the shell labelled σ, and $r_<$ and $r_>$ are respectively the lesser and greater of the two radial coordinates involved in the integral. The radial two-electron integral is found by calling the FIN... routines from routine PRNTWT.

If an N-electron target configuration ϕ_k from Eq. (7) is being considered, the result is returned from PRNTWT as a single element in VMAT(1, 1).

If the $(N+1)$-electron basis of Eq. (12) is involved, the matrix elements are returned in ((VMAT(j, j'), j = 1, ILIMIT), j' = 1, JLIMIT), where:

ILIMIT=JLIMIT=1 for bound-bound;
ILIMIT=1, JLIMIT=NRANG2 for bound-continuum;
ILIMIT=JLIMIT=NRANG2 for continuum-continuum.

SETCUP
is called from SETUP to set up channel quantum numbers in LS-coupling, given the quantum numbers of the target states and the total orbital and spin angular momentum and total parity (LRGL, NSPN, NPTY) of the $(N+1)$-electron system. It calculates the total number of coupled channels (NCHAN) and the channel orbital angular momentum l_i (array L2P), returned in common block /CROSEC/.

An angular momentum cut-off is incorporated in this routine; the largest channel angular momentum allowed is taken to be the value of LRANG2 − 1, read from the binary input file ITAPE1 from module STG1.

SETFIN
is part of the DMEL package, and extends the coupling scheme of an N-electron configuration to include a continuum electron, to return a final state continuum basis configuration in common block /MSTATE/.

SETINI
is part of the DMEL package, and extends the coupling scheme of an N-electron configuration to include a continuum electron, to return an initial state continuum basis configuration in common block /MSTATE/.

SETMX1
is the controlling routine for the evaluation, and subsequent storage on the STG2 binary output file (ITAPE3), of the $(N+1)$-electron Hamiltonian matrix elements in Eq. (14). A restart facility also allows a previously uncompleted run of STG2 to be restarted; the contents of input file ITAPE2 can be transferred to ITAPE3 under control of input parameter ICOPY (see Section 3.3 for further description).

The total wavefunction in the inner region on both sides of the matrix is expanded in the form of Eq. (12). Let φ_λ denote collectively the basis functions in Eq. (12). The inner region Hamiltonian matrix elements take the form of Eq. (14):

$$H_{\lambda\lambda'} = (\varphi_\lambda(x_1 \dots x_{N+1}) \mid H^{N+1} \mid \varphi_{\lambda'}(x_1 \dots x_{N+1}))$$

SETMX1 has three sections, corresponding to the evaluation of continuum-continuum (CC), continuum-bound (CB) and bound-bound (BB) matrix elements.

In the case of the CC matrix elements, because of symmetry only those channels with $i' < i$ are considered. Since the N-electron states are in general multi-configurational, the matrix elements are calculated as a weighted sum of contributions from each configuration in the expansion of Eq. (12). Thus:

$$\mathrm{HNP1}(j,j')_{\mathrm{CC}} = \sum_{k=1}^{\mathrm{NTCON}(i)} b_{ik} \sum_{k'=1}^{\mathrm{NTCON}(i')} b_{i'k'} (\mathcal{A}\overline{\phi}_k \tfrac{1}{r} u_{ij} \mid H^{N+1} \mid \mathcal{A}\overline{\phi}_{k'} \tfrac{1}{r} u_{i'j'}) \tag{100}$$

$$\mathrm{HNP1}(j,j')_{\mathrm{CB}} = \sum_{k=1}^{\mathrm{NTCON}(i)} b_{ik} (\mathcal{A}\overline{\phi}_k \tfrac{1}{r} u_{ij} \mid H^{N+1} \mid \chi_{j'}) \tag{101}$$

$$\mathrm{HNP1}(j,j')_{\mathrm{BB}} = (\chi_j(x_1 \dots x_{N+1}) \mid H^{N+1} \mid \chi_{j'}(x_1 \dots x_{N+1}))$$

The functions $\overline{\phi}_k$ are formed by coupling a target configuration ϕ_k from Eq. (7) with the angular and spin functions of the continuum electron to give the total angular momentum, spin and parity of the final and initial states involved. The following arrays in common block /MEDEFN/ are set before calling MATANS to evaluate the matrix element, stored in array AME, in Eqs.(97–99):

NJ,LJ defines the occupied shells;

NOSH defines the number of electrons in each shell;

J1QN defines the coupling scheme;

both for the initial and final states. When either one or two continuum orbitals are involved, the relevant elements in the array NJ are set to 999 and the array ND contains the ranges of the continuum principal quantum numbers occurring. One call of MATANS will then evaluate all matrix elements differing just in these quantum numbers. The matrix elements are accumulated in each case in the matrix HNP1. Matrix elements involving the continuum basis are calculated for each channel i in blocks with dimension NRANG2, corresponding to the number of continuum orbitals for each angular momentum.

SETMX1 takes advantage of possible duplication in the target state symmetries included; where the same target configurations ϕ_k occur in the expansion of Eq. (7) for more than one target state. Matrix elements in Eq. (100) and Eq. (101) can therefore be re-used for more than one channel i by temporarily storing the contribution from each configuration. This is done on the direct access scratch file on unit IDISC1 by calling DA2. The total matrix element is thus evaluated as a weighted sum of these configuration contributions.

SETMXR
is used in place of SETMX1 for Breit-Pauli runs. It does not have the facility for re-using matrix element contributions from the target configurations. Otherwise the operation is the same, though the loop order is different and more efficient for the case when NAST=NCFG.

SETUP
determines channel information by calling SETCUP, and writes data to output file ITAPE3.

SHRIEK
computes the factorial $n! = \mathtt{GAMMA}(n+1)$ in common block /FACT/.

SJ1QNT
is called from SETMX1 and SETMXR to set up the angular momentum quantum numbers of the atomic configuration in a form suitable for the recoupling program, leaving spaces for the continuum electron shells where appropriate.

SJ2QNT
is called from BOUND to set up the angular momentum quantum numbers of the atomic configuration in a form suitable for the recoupling program.

STG2RD
reads in the user-supplied input data for STG2. See Section 3.3 for a description of the input data. The configuration and coupling scheme data for the N-electron and $(N+1)$-electron states is generated in the CONFIG package, under control of user supplied data.

A facility in STG2RD allows a previously uncompleted run of STG2 to be restarted (ICOPY > 0); routine COPYTP is called to position of the input file ITAPE2 prior to transferring ICOPY data blocks to output file ITAPE3.

USEEAV
replaces the routine of the same name in the RKWTS package of Hibbert [50,51]. It evaluates the weights of the two-electron integrals when average energy expressions are used.

VIJOUT
is called from PNTBG2 to print out the quantum numbers and coupling schemes for each matrix element, as defined in routine SETUPE.

WRITAP
writes basic information onto the permanent output file ITAPE3 for processing by module RECUPD or STGH.

3.2. Data files

The following is a summary of the data files required by STG2. The unit numbers and file names are defined in the program. Although the variables are part of the input data (for consistency with earlier versions of RMATRX) you need only supply dummy values i.e. set them to 0. The exception is IPUNCH which should be set > 0 in the input data if you wish to write to 'CONFIG.OUT'.

IREAD = 5 — 'STG2.INP'
File type: formatted sequential input.
Written by user.
Read by routine STG2RD.
Description: N-electron and $(N+1)$-electron system data (see Section 3.3 for details).

IWRITE = 6 — 'STG2.OUT'
File type: formatted sequential output.
Written throughout STG2.
Read by user.
Description: line-printer or job output.

IPUNCH = 7 (not normally used: input as 0) — 'CONFIG.OUT'
File type: formatted sequential output.
Written by routine CONSTO in CONFIG package.
Description: configurations and coupling schemes if IPUNCH > 0 (format as in JREAD, see Section 3.3.1 for details).

IDISC1 = 11
File type: direct access scratch, record length 512 words.
Written by routine DA2, called from SETUP and SETMX1.
Read by routine DA2, called from DMEL and SETMX1.
Description: N-electron and $(N+1)$-electron state specifications and configurations from SETUP if IPOLPH = 2; $(N+1)$-electron Hamiltonian matrix elements for re-use (see description of SETMX1 in Section 3.1 for details).
Only used if insufficient space in /MEMORY/ (see &MEM processing).

IDISC2 = 0 NOT USED
IDISC3 = 0 NOT USED
IDISC4 = 0 NOT USED

ITAPE1 = 1 — 'STG1.DAT'
File type: binary sequential input.
Written by module STG1.
Read by routine CHEKTP.
Description: basic information, multipole, one-electron and bound-bound two-electron integrals (see Section 2.7 for details).

ITAPE2 = 2 (not normally used, i.e. if ICOPY = 0) — 'STG2.DMP'
File type: binary sequential input.
Written by previous run of STG2 as ITAPE3.

Read by routines AIJS, COPYTP, SETMX1, SETMXR, SETUP.
Description: ITAPE3 restart.
ITAPE3 = 3 — 'STG2H.DAT'
File type: binary sequential output.
Written by routines AIJS, COPYTP, SETMX1, SETMXR, SETUP.
Read by modules RECUPD and STGH.
Description: basic information, Hamiltonian matrices and asymptotic coefficients (see Section 3.5 for details).
ITAPE4 = 4 (used if IPOLPH = 2) — 'STG2D.DAT'
File type: binary sequential output.
Written by routine DMEL.
Read by modules RECUPD and STGH.
Description: reduced dipole matrix elements (see Section 3.6 for details).
JREAD = 8 (not normally used, i.e. if NKEY and IKEY $\neq$ 2) — 'CONFIG.INP'
File type: formatted sequential input.
Written by user, or IPUNCH, or program CIV3 [52].
Read by routine CONSTO in CONFIG package.
Description: configurations and coupling schemes if NKEY = 2. See Section 3.3.1 for details).
JDISC1 = 21 — 'RK.DAT'
File type: direct access input, record length 512 words.
Written by module STG1.
Read by routine DA2, called from CHEKTP and RDINT.
Description: bound-continuum and continuum-continuum two-electron integrals (see Section 2.8 for details).

3.3. Input data on IREAD

The user-supplied input data is read in routine STG2RD on input unit number IREAD. Free format is used, with one exception:

- FORMAT(18A4) for reading text into array TITLE.

Summary of the data records (the variable names are described in the glossary in Section 9):

1. (TITLE(K),K=1,18).
   I/O units, described in Section 3.2:
2. IWRITE, IPUNCH, IDISC1, IDISC2, IDISC3, IDISC4, ITAPE1, ITAPE2, ITAPE3, ITAPE4, JREAD, JDISC1;
   Debug parameters, described in Section 3.4:
3. IBUG1, IBUG2, IBUG3, IBUG4, IBUG5, IBUG6, IBUG7, IBUG8, IBUG9;
   Basic information:
4. ICOPY, ITOTAL, IPOLPH, (JRELOP(K), K=1,3);
5. MAXORB, NELC, NAST, NKEY, NCUT, INAST, IKEY, ICUT, NDIAG;
6. (NJCOMP(I), LJCOMP(I), I=1,MAXORB) (only if MAXORB > 0).
   For N-electron system, automatic generation of configurations:
7. (IKIP(I), I=1,NCUT) (only if NCUT > 0);
8. NOPTN (only if NKEY = 0 or 1);
9. (MNAL(I), I=1,MAXORB) (only if NKEY = 0 or ± 1 and NOPTN ≥ -1);
10. (MXAL(I), I=1,MAXORB) (only if NKEY = 0 or 1 and NOPTN $\geq$ 0);
11. (IBASSH(M,I), I=1,MAXORB), NXCITE(M) (only if NKEY = 0 or 1 and NOPTN > 0);
    repeat record 11 for M=1,NOPTN;
12. LL, LSPN, LPTY (for ground state);
    repeat record 12 for excited states to NAST (repeat records 8-12 if NKEY = 1).

If NKEY = 2 then option to read configuration data from JREAD instead of records 8–11 (Section 3.3.1).

13. If NDIAG = 0 then read configuration coefficients and eigenenergies:
    a. (NTCON(N), N=1,NAST);
    b. (AIJ(N,J), J=1,NTCON(N)), ENAT(N);
    repeat record b for N=1,NAST.
    For $(N+1)$-electron system:
14. (IKIP(I), I=1,ICUT) (only if ICUT > 0);
15. NOPTN (only if IKEY = 0 or 1);
16. (MNAL(I), I=1,MAXORB) (only if IKEY = 0 or ± 1 and NOPTN ≥ -1);
17. (MXAL(I), I=1,MAXORB) (only if IKEY = 0 or 1 and NOPTN ≥ 0);
18. (IBASSH(M,I), I=1,MAXORB), NXCITE(M) (only if IKEY = 0 or 1 and NOPTN > 0),
    repeat record 18 for M=1,NOPTN;
19. LRGL, NSPN, NPTY;
    repeat record 19 to INAST if IKEY $\neq$ 1 (repeat records 15-19 if IKEY = 1).

Important note

- Inexperienced users should set NKEY = 0, NCUT = 0, IKEY = 0, ICUT = 0, NDIAG = 1
- NAST = number of target *states*, normally. But if JRELOP(3) $\neq$ 0, then NAST = number of target *configurations*
- The target terms can be input in *any* order in record 12, providing the first term specified is the ground state. It is recommended that the terms be grouped according to their symmetry (this maximises program efficiency – see routine SETMX1), i.e. all the terms with the same symmetry should be specified consecutively in record 12. Target configurations must *always* be grouped according to their symmetry.

3.3.1. *Option: configuration input on JREAD*

If the NKEY = 2 option has been chosen, then read configuration data from JREAD instead of the recommended automatic generation in the CONFIG package. This file can be produced as output from the CONFIG package in STG2 on output IPUNCH.

Summary of the data records (the variable names are described in the glossary in Section 9):

a. NCFG
b. (NOCCSH(J), J=1,NCFG)
c. (NOCORB(I,J), I=1,NOCCSH(J))
d. (NELCSH(I,J), I=1,NOCCSH(J))
e. ((J1QNRD(I,K,J), K=1,3), I=1,2*NOCCSH(J)-1)
repeat records c–e for J=1,NCFG

3.4. *Debug prints*

Debugging prints are under the control of the IBUG parameters, specified in record 3 of the input data. Set these to zero in production runs, otherwise a large amount of output can be produced.

IBUG1 > 0 for debug printout from the two-electron angular integral routines and the associated radial integrals.
IBUG2 set in RKWTS. This is controlled by IBUG1.
IBUG3 = 1 for debug printout of the J2 and J3 coupling arrays in routines NJGRAF, J23ANG and J23SPN, and for the values of the recoupling coefficients.
IBUG4 > 1 for debug printout from HOWTS.
IBUG5 = 0 NOT USED.

IBUG6 = 1 for debug printout from AIJS, ALDAIJ and TENSOR in the evaluation of the asymptotic coefficients.

IBUG7
= 1 for printout of the N-electron configurations from the CONFIG package;
= 2 as for 1, with also printout of the $(N+1)$-electron configurations.

IBUG8
= 1 for printout of the dipole matrix elements from the DMEL package;
= 2 as for 1, with also printout from DMELBB, DMELBC, DMELCB and DMELCC.

IBUG9
= 1 for printout of the bound-bound Hamiltonian matrix elements from SETMX1 or SETMXR and asymptotic coefficients from AIJS;
= 2 as for 1, with also printout of the continuum-bound elements;
= 3 as for 2, with also printout of the continuum-continuum elements;
= 4 as for 3, with calls to PNTBG2 from routines SETMX1, SETMXR and BOUND and matrix elements printed in routine MATRX.

3.5. *Output of Hamiltonian matrices and coefficients on ITAPE3*

Summary of the output records (the variable names are described in the glossary in Section 9):

Basic data from routine WRITAP:

1. NELC, NZ, LRANG1, LRANG2, NRANG2, LAMAX, ICODE, LAM, IZESP, (JRELOP(K), K=1,3);
2. (MAXNHF(L), L=1,LRANG1), (MAXNLG(L), L=1,LRANG1), (MAXNC(L), L=1,LRANG1);
3. (EIGENS(L,J), J=1,NRANG2);
4. (ENDS(L,J), J=1,NRANG2+1);
   repeat records 3–4 for L=1,LRANG2;
5. RA, BSTO, HINT, DELTA, ETA, NIX;
6. ((COEFF(K,L), K=1,3), L=1,LRANG2);
7. N-electron matrix elements from routine BOUND for recoupling in module RECUPD, if JRELOP(3) $\neq$ 0:
   a. −JSYM;
   b. LAT(N), ISAT(N), IPTY(N), NTCO, LENGTH(N);
   c. NTCON(N), NSP, EN, (X(K), K=1,NTCON(N)) (only if NTCO < 0, i.e. JRELOP(3) = -1);
   repeat record c for | NTCO | terms;
   d. H(K), K=1,LENGTH(N);
   repeat records b–d for N=1,JSYM unique N-electron symmetries.
   Target data from routine SETUP:
8. NAST.
9. (ENAT(N), N=1,NAST), (LAT(N), N=1,NAST), (ISAT(N), N=1,NAST), (IPTY(N), N=1,NAST).
10. Configurations and spin-orbit integrals for use in RECUPD, if JRELOP(3) $\neq$ 0:
   a. NCFG, (NOCCSH(I), I=1,NCFG);
   b. NOCORB(J,I), J=1,IL), (NELCSH(J,I), J=1,IL), ((J1QNRD(J,K,I), K=1,3), J=1,2*IL−1);
   repeat record b for each target configuration I=1,NCFG;
   c. MAXORB, (NJCOMP(J), J=1,MAXORB), (LJCOMP(J), J=1,MAXORB);
   d. (NTCON(J), J=1,NAST;
   e. (NTYP(I,J), J=1,NTC), (AIJ(I,J), J=1,NTC);
   repeat record e for each target state I=1,NAST;

f. IRK5;
g. (IST1(I),I=1,LRANG1), (RSPOR1(I), I=1,IRK5;
h. IRK6;
i. (IST2(I),I=1,LRANG1), (RSPOR2(I), I=1,IRK6;
j. IRK7;
k. (RSPOR3(I,L), I=1,IRK7;
repeat record k for $l+1 \equiv$ L=2,LRANG2.

11. LRGL, NSPN, NPTY, NCFGP, IPOLPH;
12. MNP1, NCONHP, NCHAN;
13. (NCONAT(N),N=1,NAST);
14. (L2P(I),I=1,NCHAN);
15. MORE.
Hamiltonian matrix elements from routine SETMX1 or SETMXR:
16. ((HNP1(J,J'), J'=1,NRANG2), J=1,NRANG2);
repeat record 16 for all non-equivalent continuum-continuum blocks i.e. for channels I'=1,I; I=1,NCHAN, the upper triangle of blocks are written out by rows;
17. ((HNP1(J,J'), J'=1,NCFGP), J=1,NRANG2);
repeat record 17 for all bound-continuum blocks i.e. for channel I=1,NCHAN;
18. (HNP1(J,J'), J'=J,NCFGP);
repeat record 18 for all bound-bound rows, i.e. for J=1,NCFGP.
Asymptotic coefficients from routine AIJS:
19. ((CF(I,I',K), K=1,LAMAX), I'=I,NCHAN);
repeat record 19 for I=1,NCHAN.
Repeat records 11-19 if MORE>0.

3.6. *Output of dipole matrix elements on ITAPE4*

Summary of the output records (the variable names are described in the glossary in Section 9):
1. ((DEL(J',J), J=1,NRANG2), J'=1,NRANG2);
2. ((DEV(J',J), J=1,NRANG2), J'=1,NRANG2);
repeat records 1-2 for each initial state channel;
3. ((DEL(J',J), J=1,NRANG2), J'=1,MCFGP);
4. ((DEV(J',J), J=1,NRANG2), J'=1,MCFGP);
5. ((ABUTL(J,I'), J=1,NRANG2), I'=1,MCHAN);
6. ((ABUTV(J,I'), J=1,NRANG2), I'=1,MCHAN);
repeat records 1-6 for each final state channel;
7. ((DEL(J,J'), J=1,NCFGP), J'=1,NRANG2);
8. ((DEV(J,J'), J=1,NCFGP), J'=1,NRANG2);
repeat records 7-8 for each initial state channel;
9. ((DEL(J',J), J=1,NCFGP), J'=1,MCFGP);
10. ((DEV(J',J), J=1,NCFGP), J'=1,MCFGP);
repeat records 9-10 if NCFGP exceeds the dimension of the DEL and DEV arrays;
11. ((ABUTL(J,I'), J=1,NCFGP), I'=1,MCHAN);
12. ((ABUTV(J,I'), J=1,NCFGP), I'=1,MCHAN);
13. ((ABUTL(I,J'), I=1,NCHAN), J'=1,NRANG2);
14. ((ABUTV(I,J'), I=1,NCHAN), J'=1,NRANG2);
repeat records 13-14 for each initial state channel;
15. ((ABUTL(J',I), I=1,NCHAN), J'=1,MCFGP);

```
 1  MNRECU AARECU BOUNDJ ----
 9               COPYTP RECOV2
10               DJZERO DA2
11                      RECOV2
12               DMOUT  DA2
13                      RECOV2
14               HJZERO DA2
15                      RECOV2
16               IRECUP FACTT(*)
17               NJCHAN NUMSYM LSJTRI
18                      RECOV2
19               READS  RECOV2
20               RECUD  ----
27               RECUPJ ----
38               SPINOR ----
57               WRIT3  DA2
58                      RECOV2
59               WRITAP
```

Fig. 11. Calling tree for module RECUPD. The routines are given in alphabetical order within each branch **not** the order in which they are actually called. The routines marked with (*) are in module STGLIB.

16. ((ABUTV(J′,I), I=1,NCHAN), J′=1,MCFGP);
17. ((ABUTL(I,I′), I=1,NCHAN), I′=1,MCHAN);
18. ((ABUTV(I,I′), I=1,NCHAN), I′=1,MCHAN);
19. MAXM1,(CGC(L),L=1,MAXM1);
20. ((AC(I,I′), I′=1,MCHAN), I=1,NCHAN);
21. ((BLC(I,I′), I′=1,MCHAN), I=1,NCHAN);
22. ((BVC(I,I′), I′=1,MCHAN), I=1,NCHAN).

4. Module RECUPD

This module is an optional stage in RMATRX1. It includes relativistic effects in the Breit-Pauli approximation, and is run after module STG2 and before STGH.

RECUPD must be linked with the STGLIB module in order to form an executable program. Routines DRACAH, FACTT, HSLDR, SETUPE and TENSOR plus routines which they call are obtained from STGLIB (see Section 7).

Figs. 11–15 display a flow diagram for the routines in RECUPD.

There are four main computational sections in RECUPD (the controlling routines are named in brackets):

- initialisation and reading input files, (READS, COPYTP, IRECUP);
- options to diagonalize the target Hamiltonian and define term-coupling coefficients, and output (BOUNDJ, WRITAP);
- recoupling and output of $(N+1)$-electron Hamiltonian matrices and long-range potential coefficients (NJCHAN, HJZERO, RECUPJ, SPINOR, WRIT3);
- recoupling and output of dipole matrices, if any (DJZERO, RECUD, DMOUT).

4.1. Outline of RECUPD calculation

The purpose of RECUPD is to transform Hamiltonian matrices, long-range potential coefficients and (optionally) dipole matrices, defined by Eq. (14), Eq. (29) and Eq. (48), from *LS*- to pair-coupling by means of a unitary transformation.

```
 1      BOUNDJ BSPNO  DRACAH(*)
 2                    LSJTRI
 3                    SPINBB FINBBR
 4                           SETUPE(*)
 5                           TENSOR(*)
 6             HSLDR(*)
 7             LSJTRI
 8             RECOV2
```

Fig. 12. Calling tree for module RECUPD, BOUNDJ section

```
20      RECUD  DAFILA DA2
21                    RECOV2
22             DFIND  RECOV2
23             DMES   DA2
24                    RECOV2
25             DRACAH(*)
26             LSJTRI
```

Fig. 13. Calling tree for module RECUPD, RECUD section

```
27      RECUPJ DA2
28             DAFILA > 20
29             DEGEN
30             HFIND  RECOV2
31             LSCONT DA2
32             LSJCUP JLRC   DRACAH(*)
33                    LSJTRI
34                    RECOV2
35             LSJTRI
36             NDEGEN DA2
37             RECOV2
```

Fig. 14. Calling tree for module RECUPD, RECUPJ section

```
38      SPINOR DA2
39             DAFILA > 20
40             DRACAH(*)
41             LSJTRI
42             SPINBB > 3
43             SPINBC FINBCR
44                    SETR
45                    SETUPE(*)
46                    TENSOR(*)
47             SPINCB FINBCR
48                    SETL
49                    SETUPE(*)
50                    TENSOR(*)
51             SPINCC FINBBR
52                    FINCCR
53                    SETL
54                    SETR
55                    SETUPE(*)
56                    TENSOR(*)
```

Fig. 15. Calling tree for module RECUPD, SPINOR section

Each total angular momentum and parity symmetry of the $(N+1)$-electron system is considered individually.

If the spin-orbit operator is to be included explicitly, then STG2 must be run with each state represented by a single configuration. The corresponding LS-coupled Hamiltonian matrices and long-range potential coefficients are read in from ITAPE2 and transformed to pair-coupling. Matrix elements of the spin-orbit operator are then calculated in LSJ-coupling, transformed to pair-coupling and added to the transformed STG2 matrices.

If the spin-orbit interaction is not explicitly included then STG2 should be run in the normal LS-coupled way subject to the condition that NAST = number of N-electron configurations generated. The Hamiltonian matrices and long-range potential coefficients are then recoupled using term-coupling coefficients. The diagonal elements of the continuum-continuum Hamiltonian matrix are then adjusted to reproduce the observed energy splittings of the target atom.

The Hamiltonian matrices and long-range potential coefficients are then stored in unformatted form on an output file (ITAPE3). This can be read as input by the module STGH, enabling the calculation of cross sections between fine-structure levels both above and below threshold.

Similar transformations are then applied to the dipole matrices (if present), using recoupling coefficients between the LS- and pair-coupling schemes already computed in the transformation of the Hamiltonian matrices. Considering the new coupling scheme, all possible initial and final symmetries in this scheme are looped over. Whenever a dipole transition is deemed possible then pairs of LS-symmetries are considered (one contributing to the recoupling for the initial state wavefunction, the other contributing to the recoupling for the final state wavefunction) until a pair is found linked by a dipole transition in the LS-coupling scheme. The corresponding dipole matrix elements are read from the STG2 input file ITAPE1, and their contribution to the dipole matrix under construction in the new coupling scheme is calculated. This is repeated for all sets of dipole matrix elements in the LS-coupling scheme that can be found to contribute. The newly formed dipole matrix is then written out to output file ITAPE4, and the process repeated for any further dipole transitions.

For completeness we now define the main equations that are required in the reformulation of the R-matrix theory in terms of the Breit-Pauli Hamiltonian. Readers are referred to Scott and Burke [71] for a complete description of the equations and notation.

The Breit-Pauli Hamiltonian (both $H_{\rm BP}^{N}$ and $H_{\rm BP}^{N+1}$) separates into non-fine-structure terms ($H_{\rm nfs}$) and terms involving the spin-orbit interaction ($H_{\rm SO}$) as in Eq. (59) and Eq. (63),

$$H_{\rm BP} = H_{\rm nfs} + H_{\rm SO}$$

The target eigenstates Φ_i are expanded in terms of a single configuration basis as in Eq. (7), though in the intermediate-coupling scheme Eq. (64) for each target level angular momentum and parity, such that the expansion coefficients c_{ij} are chosen to ensure that

$$\langle \Phi_i \mid H_{\rm BP}^{N} \mid \Phi_j \rangle = E_i^N \delta_{ij} \tag{102}$$

The target Hamiltonian matrix can be set up and diagonalized within RECUPD, in routine BOUNDJ, or the coefficients c_{ij} and energy levels E_i^N can be supplied as input to RECUPD by the user.

The $(N+1)$-electron R-matrix basis functions ψ_k are defined as in Eq. (13), again using the intermediate-coupling scheme, for each total angular momentum (J) and parity, i.e. the conserved quantum numbers are now J, M_J and parity.

Let φ_λ denote collectively the basis functions in Eq. (12), for a given total $J\pi$-symmetry (cf. Eq. (14)). Transforming the $(N+1)$-electron Hamiltonian matrix requires the evaluation of matrix elements of the type

$$(\varphi_\lambda \mid H_{\rm BP}^{N+1} \mid \varphi_{\lambda'}) \tag{103}$$

These matrix elements can be divided into three types: continuum-continuum, bound-continuum and bound-bound. The long-range potential coefficients can also be transformed in a similar fashion. Each matrix block is most conveniently calculated in LSJ-coupling and then transformed to the appropriate coupling scheme by means of a unitary transformation.

1. Continuum-continuum Hamiltonian matrix elements

$$\begin{aligned}(\Delta_i(J_il_i)K_i\tfrac{1}{2};J\pi\,|H_{\rm BP}^{N+1}|\,\Delta_j(J_jl_j)K_j\tfrac{1}{2};J\pi) &= \sum_{LL'SS'C_iL_iS_iC_jL_jS_j} C(\Delta_iJ_i;C_iL_iS_il_iK_i\tfrac{1}{2}LS;J\pi)C(\Delta_jJ_j;C_jL_jS_jl_jK_j\tfrac{1}{2}L'S';J\pi)\\ &\times[\ (C_i(L_il_i)L(S_i\tfrac{1}{2})S;J\pi\ |\ H_{\rm SO}^{N+1}\ |\ C_j(L_jl_j)L'(S_j\tfrac{1}{2})S';J\pi)\\ &+\delta_{LL'}\,\delta_{SS'}\ (C_i(L_il_i)L(S_i\tfrac{1}{2})S\pi\ |\ H_{\rm nfs}^{N+1}\ |\ C_j(L_jl_j)L(S_j\tfrac{1}{2})S\pi)\ ]\end{aligned} \tag{104}$$

where

$$C(\Delta_iJ_i;C_iL_iS_il_iK_i\tfrac{1}{2}LS;J\pi) = B^{J\pi}(\Delta_iJ_i;C_iL_iS_i)\times \text{COEF} \tag{105}$$

and

$$\text{COEF} = \sqrt{(2J_i+1)(2L+1)(2K_i+1)(2S+1)}\,W(Ll_iS_iJ_i;L_iK_i)\ W(LJS_i\tfrac{1}{2};SK_i)$$

2. Bound-continuum Hamiltonian matrix elements

$$\begin{aligned}(LS;J\pi\,|H_{\rm BP}^{N+1}|\,\Delta_i(J_il_i)K_i\tfrac{1}{2};J\pi) = \sum_{L'S'C_iL_iS_i} & C(\Delta_iJ_i;C_iL_iS_il_iK_i\tfrac{1}{2}L'S';J\pi)\\ &\times[\ (LS;J\pi\ |\ H_{\rm SO}^{N+1}\ |\ C_i(L_il_i)L'(S_i\tfrac{1}{2})S';J\pi)\\ &+\delta_{LL'}\,\delta_{SS'}\ (LS\pi\ |\ H_{\rm nfs}^{N+1}\ |\ C_i(L_il_i)L(S_i\tfrac{1}{2})S\pi)\ ]\end{aligned} \tag{106}$$

3. Bound-bound Hamiltonian matrix elements

$$\begin{aligned}(LS;J\pi\ |\ H_{\rm BP}^{N+1}\ |\ L'S';J\pi) &= (LS;J\pi\ |\ H_{\rm SO}^{N+1}\ |\ L'S';J\pi)\\ &+\delta_{LL'}\,\delta_{SS'}\ (LS\pi\ |\ H_{\rm nfs}^{N+1}\ |\ LS\pi)\end{aligned} \tag{107}$$

4. The long-range potential coefficients

$$\begin{aligned}a_{ij}^{\lambda} &= \left\langle \Delta_i(J_il_i)K_i\tfrac{1}{2};J\pi\ \middle|\ \sum_{n=1}^{N} r_n^{\lambda}P_{\lambda}(\cos\theta_{n,N+1})\ \middle|\ \Delta_j(J_jl_j)K_j\tfrac{1}{2};J\pi\right\rangle\\ &= \sum_{LSC_iL_iS_iC_jL_jS_j} C(\Delta_iJ_i;C_iL_iS_il_iK_i\tfrac{1}{2}LS;J\pi)C(\Delta_jJ_j;C_jL_jS_jl_jK_j\tfrac{1}{2}L'S';J\pi)\\ &\times\left\langle C_i(L_il_i)L(S_i\tfrac{1}{2})S;J\pi\ \middle|\ \sum_{n=1}^{N} r_n^{\lambda}P_{\lambda}(\cos\theta_{n,N+1})\ \middle|\ C_j(L_jl_j)L'(S_j\tfrac{1}{2})S';J\pi\right\rangle\end{aligned} \tag{108}$$

It can be shown that the continuum-continuum matrix may be split into the sum of two matrices, since $H^{N+1} = H^N + H(N+1)$, where $H(N+1)$ are the one- and two-electron interactions with the $(N+1)$th electron. The continuum-continuum matrix associated with H^N is clearly diagonal and contains the eigenenergies of the target atom. This allows us to adjust the diagonal elements of Eq. (104) to reproduce the observed energy splittings of the target atom. In this way we effectively account for the two-electron terms of the Breit-Pauli Hamiltonian excluded from Eq. (59) (see Eq. (4) of Scott and Burke [71]), and correctly account for the kinematics of the continuum electron in the external region.

The above approach is valid provided the relativistic two-electron terms are small compared with the relativistic one-electron terms, as will be the case as the nuclear charge increases. However, in lighter systems this may not be true, and an alternative procedure may be adopted where the dominant effects result from correlation and intermediate-coupling in the target atom. Here we expand the Φ_i of Eq. (102) in terms of a CI basis ϕ_j which diagonalizes the non-relativistic $H^N_{\rm nr}$:

$$\Phi_i = \sum_j f_{ij}\phi_j \tag{109}$$

The expansion coefficients f_{ij} are now the term-coupling coefficients of Jones [56] evaluated using the full Breit-Pauli Hamiltonian defined in Eq. (4) of Scott and Burke [71]. We now approximate Eq. (103) by

$$(\varphi_\lambda \mid H^{N+1}_{\rm nr} \mid \varphi_{\lambda'}) \tag{110}$$

again adjusting the diagonal elements of the continuum-continuum matrix to effectively account for the operators excluded from the Hamiltonian. The corresponding matrix elements are given by Eq. (104), Eq. (106) and Eq. (108), with C_i replaced by Γ_i (as defined by Jones [56]) and $H^{N+1}_{\rm SO}$, $H^{N+1}_{\rm mass}$, $H^{N+1}_{\rm D_1}$ removed from the right hand side, and where Eq. (105) becomes

$$C(\Delta_i J_i; \Gamma_i L_i S_i l_i K_i \tfrac{1}{2} LS; J\pi) = F^{J\pi}(\Delta_i J_i; \Gamma_i L_i S_i) \times \texttt{COEF} \tag{111}$$

This approach is similar to that of Saraph [69,70], except that it correctly accounts for the kinematics of the continuum electron in the outer region, and allows a consistent treatment both above and below threshold. From now on we will write the equations in terms of C_i. The corresponding ones for Γ_i may be trivially obtained.

4.2. Routines

MNRECU
is the program routine and contains all COMMON blocks used in RECUPD. It sets /MEMORY/ pointers and calls AARECU.

AARECU
is called by MNRECU and controls the RECUPD computation, invoking the four main computational sections as summarised above.

There is a loop over the required $(N+1)$-electron system $J\pi$-symmetries to recouple Hamiltonian matrices and long-range potential coefficients, which is completed before entering loops over $(J\pi)_f$ and $(J\pi)_i$, the final and initial symmetries, to recouple any dipole matrices.

BOUNDJ
calculates energies and CI coefficients of fine-structure N-electron states, by recoupling the appropriate LS-coupled target Hamiltonians (stored in /BOUNDD/ by routine COPYTP) and diagonalizing as in Eq. (102).

As an option, term-coupling coefficients are computed and written onto `IPUNCH` (see Section 4.6) for transforming K-matrices in LS-coupling. This procedure requires the eigenvectors of the non-relativistic target Hamiltonian from module STG2, which become available only if STG2 has been run with the specification `JRELOP(3)` $= -1$.

BSPNO
is similar to SPINOR, but is called by routine BOUNDJ to add in the spin-orbit contribution to fine-structure target Hamiltonians.

COPYTP
reads the header data on the file ITAPE2 from module STG2, and positions file at start of the $LS\pi$-dependent data (i.e. at record 11, see Section 3.5).

DA2
is used both in the temporary storage of channel recoupling data (IDISC1), and in the temporary storage of J-coupled continuum-continuum matrix elements (IDISC2). See the description of this routine in Section 2.

DAFILA
writes or retrieves channel recoupling data involved in recoupling a given $SL\pi$ to $J\pi$.

Data are held in /MEMORY/ if sufficient space (&MEM) is available, otherwise a direct access scratch file IDISC1 is opened for temporary storage. Pointers in IRECA are set positive for /MEMORY/ location, negative for file record location.

DEGEN
is called if the spin-orbit interaction is not explicitly included (JRELOP(3) = 0). In this case term-coupling coefficients are being used. If we were to transform the LS-coupled Hamiltonian matrices using term coupling coefficients spurious off-diagonal elements would occur in the transformed matrices. This is because the N-electron wavefunctions diagonalize $H^N_{\rm BP}$ not $H^N_{\rm nr}$ (see Eq. (110)). To bypass this obstacle routine DEGEN adjusts the diagonal elements of the continuum-continuum LS-coupled Hamiltonian matrices (stored in array HLS) so that the target energy levels are degenerate with the ground state. The equations are then transformed by routine RECUPJ, and NDEGEN is called to readjust the diagonal elements to give the observed target splittings.

DFIND
is called from RECUD to read the next set of dipole matrix elements on the input file (ITAPE1) from module STG2, for a given pair of LS-symmetries linked by a dipole transition.

DJZERO
initialises various two-dimensional arrays that will be used later to hold continuum-continuum, bound-continuum, continuum-bound and bound-bound dipole matrix elements in the new coupling scheme, as they are built up from transformed contributions evaluated in the original LS scheme.

If there is insufficient space in /MEMORY/, DJZERO will initialise scratch file IDISC2 for overflow.

DMES
is called by RECUD to work out the contribution each set of LS-coupled dipole matrix elements makes to the desired dipole matrix in the $J\pi$-coupling scheme. The contributing set of dipole matrix elements in LS-coupling are input, as are the Racah coefficient and phase factor needed for the transformation. Here we are required to determine reduced matrix elements of the form (cf. Eq. (48))

$$D_{\lambda\lambda'} = (\varphi_\lambda \| M_1 \| \varphi_{\lambda'})$$

where M_1 is a tensor of rank 1 representing the electric dipole operator. Due to the fact that the total angular momentum and parity are not the same in the initial and final states, there are four types of matrix element to be considered: continuum-continuum, bound-continuum, continuum-bound and bound-bound.

The factor

$$\mathrm{RAC} = \sqrt{(2J+1)(2J'+1)}\,(-1)^{L'+S-J-1}\,W(J'L'JL;S1)$$

is input to the routine.

1. Continuum-continuum dipole matrix elements

$$(\Delta_i(J_il_i)K_i\tfrac{1}{2};J\pi\|M_1\|\Delta_j(J_jl_j)K_j\tfrac{1}{2};J'\pi') = \sum_{LL'SC_iL_iS_iC_jL_jS_j} \mathrm{RAC} \times C(\Delta_iJ_i;C_iL_iS_il_iK_i\tfrac{1}{2}LS;J\pi)C(\Delta_jJ_j;C_jL_jS_jl_jK_j\tfrac{1}{2}L'S;J'\pi') \times (C_i(L_il_i)L(S_i\tfrac{1}{2})S\pi\|M_1\|C_j(L_jl_j)L'(S_j\tfrac{1}{2})S\pi') \tag{112}$$

For each of the two symmetries each coupled channel is examined to see if it will obtain a contribution from the channels in the two *LS*-symmetries being considered. Information previously stored during the transformation of the Hamiltonian matrices allows this to be decided readily. If a channel coupled to the new initial state symmetry has a counterpart in the *LS*-symmetry and there is a similar correspondence between the final state symmetries, then the particular continuum-continuum block of the *LS*-coupled dipole matrix corresponding to these channels makes some non-zero contribution to the corresponding channel block in the new scheme. The size of the contribution is determined by the recoupling coefficients etc. previously calculated, and multiplied in at this stage.

2. Continuum-bound dipole matrix elements

$$(LS;J\pi\|M_1\|\Delta_i(J_il_i)K_i\tfrac{1}{2};J'\pi') = \sum_{L'C_iL_iS_i} \mathrm{RAC} \times C(\Delta_iJ_i;C_iL_iS_il_iK_i\tfrac{1}{2}L'S;J'\pi') \times (LS\pi\|M_1\|C_i(L_il_i)L'(S_i\tfrac{1}{2})S\pi') \tag{113}$$

Here each channel coupled in the final state alone is considered. From a treatment of the continuum-continuum blocks above a knowledge of the correspondence between channels in the two coupling schemes has been retained and the possible contribution from the *LS*-coupled continuum-bound dipole matrix elements multiplied by the recoupling coefficient etc. is taken.

3. Bound-continuum dipole matrix elements

$$(\Delta_i(J_il_i)K_i\tfrac{1}{2};J\pi\|M_1\|L'S;J'\pi') = \sum_{LC_iL_iS_i} \mathrm{RAC} \times C(\Delta_iJ_i;C_iL_iS_il_iK_i\tfrac{1}{2}LS;J\pi) \times (C_i(L_il_i)L(S_i\tfrac{1}{2})S\pi\|M_1\|L'S\pi') \tag{114}$$

This is similar in treatment to the continuum-bound case above except that now the continuum part is associated with the initial state and the bound part with the final state.

4. Bound-bound dipole matrix elements

$$(LS;J\pi\|M_1\|L'S;J'\pi') = \mathrm{RAC} \times (LS\pi\|M_1\|L'S\pi') \tag{115}$$

Each bound-bound dipole matrix element in the *LS* scheme is simply carried over into the new coupling scheme and new dipole matrix, albeit multiplied by a phase factor and Racah coefficient.

DMOUT
writes out to output file ITAPE4 blocks of dipole matrix elements formed in the new coupling scheme.

FINBBR,FINBCR,FINCCR
find a bound-bound(BB), bound-continuum(BC) or continuum-continuum(CC) one-electron, spin-orbit integral from the arrays RSPOR1, RSPOR2 or RSPOR3 respectively.

HFIND
reads an *LS*-coupled Hamiltonian matrix and associated long-range potential coeficients from the STG2 file (ITAPE2).

HJZERO
initialises the scratch file (`IDISC1`) and various two-dimensional arrays that will be used later to hold continuum-continuum, bound-continuum and bound-bound Hamiltonian matrix elements and long-range potential coefficients in the new coupling scheme, as they are built up from transformed contributions evaluated in the original LS scheme.

IRECUP
is called once from AARECU for initialisation:
- calls STGLIB library routine FACTT to define logs of factorials in /FACTS/, these are needed for the DRACAH routine;
- partitions /MEMORY/ between channel recoupling data and continuum-continuum matrix elements.

JLRC
evaluates the recoupling coefficient which provides the transformation from LSJ to J_il coupling, used in Eq. (105) and Eq. (111).

$$\mathtt{COEF} = \sqrt{(2J_i+1)(2L+1)(2S+1)(2K_i+1)}\; W(LJS_i\tfrac{1}{2};SK)\; W(Ll_iS_iJ_i;L_iK_i)$$

The angular momentum quantum numbers in the argument list have the following correspondence: (`J1, J2, J3, J4, J5, J6, J7, J8, J9`) $\equiv (2L_i, 2S_i, 2J_i, 2K_i, 2l_i, 2L, 2S, 1, 2J)$

LSCONT
applies the contributions, described in Eqs.(104–108), to the non-fine-structure Hamiltonian matrices and long-range potential coefficients, which arise from the current LS-symmetry being considered (`LRGL, NSPN`).

LSJCUP
calculates the channel recoupling contributions from this LS-symmetry to the current J-symmetry.

LSJTRI
is an integer function used to check the parity and triangular relations between L, S and J.

NDEGEN
is called if `JRELOP(3)` = 0. This routine adjusts the diagonal elements of the transformed continuum-continuum Hamiltonian matrices so that the energy levels in the target have the observed splittings. See routine DEGEN for further details.

NJCHAN
for a given $J\pi$-symmetry, this routine defines and stores the channels in the pair-coupling scheme. It then calls routine NUMSYM to determine the number of $LS\pi$-symmetries required for convergence in Eq. (104), Eq. (106) and Eq. (108).

NUMSYM
determines the number of $SL\pi$-symmetries required in the transformation for convergence (see Eq. (104), Eq. (106) and Eq. (108)) for the J value under consideration. It uses the condition

$$|S_i^{\min} - \tfrac{1}{2}| \le S \le |S_i^{\max} + \tfrac{1}{2}|$$

This gives all the possible S values, where $S_i^{\min}$ and $S_i^{\max}$ are the minimum and maximum spins of the target states in module STG2. For each S value the possible L values are given by

$$|\, J - S \,| \le L \le |\, J + S \,|$$

For each $LS\pi$-symmetry it then checks if there are any channels coupled in. The total number of symmetries required is held in ICOUNT. If not enough symmetries have been provided by STG2 the code will terminate in routine RECUPJ with an appropriate message.

READS
reads user input data from unit number IREAD as described in Section 4.4.

RECOV2
See the description of this routine in Section 2.

RECUD
is the controlling routine for the transformation of dipole matrices. It loops over pairs of LS-symmetries in the order in which they are stored on the input file (ITAPE1) from module STG2, and calls routine DFIND to read the LS-coupled dipole matrix. The parity and triangular relations between L, S and J are checked by calling function LSJTRI. If this particular LS-coupled dipole matrix contributes to the current J-coupled dipole matrix, then: routine DAFILA is called to retrieve channel recoupling data calculated previously by routine RECUPJ; and routine DMES is called to update the J-coupled dipole matrix.

RECUPJ
is the controlling routine for the transformation of Hamiltonian matrices. Each LS-coupled Hamiltonian matrix is read in turn from the input file ITAPE2, as written by module STG2, and the parity and triangular relations between L, S and J are checked by calling function LSJTRI. Before applying the transformations, routine LSJCUP is called to determine the necessary recoupling information for all channels and stores this information in arrays NTERM, ICHAN and PV. If JRELOP(3) = 1 or a photoionization calculation, these arrays are also written out to scratch file (IDISC1), for later use in routines SPINOR or RECUD. The transformation is carried out in routine LSCONT.

SETL
extends the coupling scheme of an N-electron configuration to include a continuum electron on the left hand side of a matrix element.

SETR
extends the coupling scheme of an N-electron configuration to include a continuum electron on the right hand side of a matrix element.

SPINBB
evaluates the bound-bound spin-orbit matrix elements which arise from the $(N+1)$-electron configurations associated with symmetries (LRGL, NSPN) and (LLRGL, NNSPN). If a diagonal block is being considered only the upper triangle is calculated.

SPINBC
evaluates the bound-continuum spin-orbit matrix elements where the bound terms arise from the $(N+1)$-electron configurations associated with the symmetry (LRGL, NSPN) and the continuum terms are associated with the symmetry (LLRGL, NNSPN). If a diagonal block is being considered only the upper triangle is calculated.

SPINCB
evaluates the continuum-bound spin-orbit matrix elements where the continuum terms are associated with the symmetry (LRGL, NSPN), and the bound terms arise from the $(N+1)$-electron configurations associated with the symmetry (LLRGL, NNSPN). Note that we use the fact that

$$\begin{aligned}&\langle LS; J\pi \,|H_{\mathrm{SO}}^{N+1}|\, C_i(L_il_i)L'(S_i\tfrac{1}{2})S'; J\pi\rangle\\ &\quad = \langle C_i(L_il_i)L'(S_i\tfrac{1}{2})S'; J\pi \,|H_{\mathrm{SO}}^{N+1}|\, LS; J\pi\rangle\end{aligned}$$

This routine is not called if a diagonal block is being considered.

SPINCC
evaluates the continuum-continuum spin-orbit matrix elements which arise from the continuum terms associated with the symmetries (LRGL, NSPN) and (LLRGL, NNSPN). If a diagonal block is being considered only the lower triangle is calculated.

SPINOR
is the controlling routine for the evaluation of the spin-orbit matrix elements required in Eq. (104), Eq. (106) and Eq. (107).

The spin-orbit matrix elements are not diagonal in L and S, therefore all combinations of L, S, L', S' are considered. Since the total matrix is symmetric about the diagonal only the lower triangle of blocks is explicitly calculated, the upper triangle being the transpose of the lower. For each set of (LRGL, NSPN) and (LLRGL, NNSPN) symmetries, routines SPINBB, SPINBC, SPINCB and SPINCC are called to evaluate the spin-orbit matrix elements in LSJ-coupling. These are then transformed to pair-coupling and added to the transformed Hamiltonian matrices calculated in routine RECUPJ.

Certain symmetry properties are used to cut down the amount of computation. See the comments in the source listing for details.

All the spin-orbit matrix elements are evaluated in LSJ-coupling according to the expression given in Eq. (28) of Glass and Hibbert [47].

$$\left\langle \varphi \left| \frac{\alpha^2}{2} \sum_n (l_1(n).s_1(n))\ \zeta(r_n) \right| \varphi' \right\rangle = (-1)^{L+S'-J}\ W(LL'SS';1J)$$
$$\times \frac{\alpha^2}{2}\ \text{VSHELL}\ \left[\tfrac{3}{2} l_\rho(l_\rho+1)(2l_\rho+1)\right]^{1/2}$$
$$\times \delta_{l_\rho l_\sigma} \left(U_{n_\rho l_\rho} \left| \frac{Z}{r^3} \right| U_{n_\sigma l_\sigma} \right)$$

where ρ and σ are the 'interacting' subshells. $W(LL'SS';1J)$ = RAC is a Racah coefficient from a call to DRACAH, and is passed as an argument to the SPIN... routines, which call TENSOR for VSHELL, and the FIN...R routines for the spin-orbit radial integrals.

WRIT3
writes out records to the output file ITAPE3, containing the recoupled Hamiltonian matrices and long-range potential coefficients. See Section 4.7 for details.

WRITAP
writes basic information as a header onto the output file ITAPE3, for processing by module STGH. See Section 4.7 for details.

4.3. Data files

The following is a summary of the data files required by RECUPD. The unit numbers and file names are defined in the program. Although the variables are part of the input data (for consistency with earlier versions of RMATRX) you need only supply dummy values i.e. set them to 0. The exceptions are: IPUNCH which should be set > 0 in the input data if you wish to write to 'TCC.DAT'; ITAPE1 which should be set > 0 in the input data if you wish to read from 'STG2D.DAT' and write to 'RECUPD.DAT', i.e. handle dipole data; and ITAPE3 which should be set > 0 in the input data if you wish to write to 'RECUPH.DAT'.

IREAD = 5 — 'RECUPD.INP'
File type: formatted sequential input.
Written by user.
Read by routine READS.
Description: N-electron and $(N+1)$-electron system data (see Section 4.4 for details).

IWRITE = 6 — 'RECUPD.OUT'
File type: formatted sequential output.
Written throughout RECUPD.
Read by user.
Description: line-printer or standard output — the log file.

IPUNCH = 7 (not normally used: input as 0) — 'TCC.DAT'
File type: formatted sequential output.
Written by BOUNDJ.
Read by program JAJOM (Saraph [69,70]).
Description: term-coupling coefficients, see Section 4.6.
(used only if IPUNCH > 0).

IDISC1 = 11
File type: direct access binary scratch, record length 512 words.
Written by routine DA2, called from DAFILA.
Read by routine DA2, called from DAFILA
Description: channel recoupling data
(only used if JRELOP(3) = 1 or photoionization run if insufficient memory).

IDISC2 = 12
File type: direct access binary scratch, record length 512 words.
Written by routine DA2, called from various routines.
Read by routine DA2, called from various routines.
Description: recoupled continuum-continuum matrix elements (used only if insufficient memory).

ITAPE1 = 1 (input as 0 if no dipole matrices) — 'STG2D.DAT'
File type: binary sequential input.
Written by module STG2
Read by routine DFIND
Description: LS-coupled dipole matrix elements
(only used in photoionization run — see Section 3.6 for details).

ITAPE2 = 2 — 'STG2H.DAT'
File type: binary sequential input.
Written by module STG2.
Read by routines COPYTP.
Description: basic information, LS-coupled Hamiltonian matrices.
(see Section 3.5 for details)

ITAPE3 = 3 — 'RECUPH.DAT'
File type: binary sequential output.
Written by routines WRITAP and WRIT3.
Read by module STGH.
Description: basic information, recoupled Hamiltonian matrices.
(see Section 4.7 for details).

ITAPE4 = 4 (not used if ITAPE1 = 0 in input data) — 'RECUPD.DAT'
File type: binary sequential output.
Written by routine DMOUT.

Read by module STGH.
Description: recoupled dipole matrices
(see Section 4.8 for details).

4.4. Input data on IREAD

The user-supplied input data is read in routine READS on input unit number `IREAD`. Free format is used, with one exception:

- FORMAT(18A4) for reading text into array `TITLE`.

Summary of the data records (the variable names are described in the glossary in Section 9):

1. `(TITLE(K),K=1,18)`.
   I/O units, described in Section 4.3:
2. `IWRITE, IPUNCH, IDISC1, IDISC2, ITAPE1, ITAPE2, ITAPE3, ITAPE4.`
   Debug parameters, described in Section 4.5:
3. `IBUG1, IBUG2, IBUG3, IBUG4, IBUG5, IBUG6, IBUG7, IBUG8, IBUG9.`
   N-electron target data:
4. `JNAST, ICHECK, IPHOT;`
5. `(JJ(I),I=1,JNAST);`
6. `(JPTY(I),I=1,JNAST);`
7. if `ICHECK` = 0 then read target data:
   a. `(ENAT(I),I=1,JNAST),`
   b. `(JNTCON(I),I=1,JNAST),`
   c. `(B(I,K),K=1,JNTCON(I)),`
   repeat record c for `I=1,JNAST,`
   d. `(LSVALU(I,K),K=1,JNTCON(I))`
   repeat record d for `I=1,JNAST.`
   $(N+1)$-electron symmetries:
8. `IJNAST;`
9. `J2(J), JP(J);`
   repeat record 9 for `J=1,IJNAST.`

Important note

- The inexperienced user should run STG2 with `NDIAG` = 1 and RECUPD with `ICHECK` = 1; thus the target energies and eigenvectors will be automatically generated by a diagonalization of the Breit-Pauli Hamiltonian in routine BOUNDJ.
- If `ICHECK` = 0, then level energies `ENAT` must be supplied as input:
  `JRELOP(3)` = 0 the energies should give the observed splittings,
  `JRELOP(3)` = 1 the energies should be the theoretical ones.

4.5. Debug prints

Debugging prints are under the control of the `IBUG` parameters, specified in record 3 of the input data. Set these to zero in production runs, otherwise a large amount of output can be printed.

`IBUG1`
= 0 normally;
= 1 printout from routine CFP.

IBUG2
= 0 normally;
= 1 printout from routine NJGRAF.
IBUG3 = 0 NOT USED.
IBUG4
= 0 normally;
= 1 for a printout of the arrays defining the channels from routine NJCHAN.
IBUG5
= 0 normally;
= 1 for a printout from routines LSCONT and JLRC of the arrays containing the recoupling information for all the channels;
= 2 as for 1, together with a printout from routine RECUPJ of the transformed bound-bound Hamiltonian matrix blocks;
= 3 as for 2, together with a printout from routine RECUPJ of the transformed bound-continuum matrix blocks;
= 4 as for 3, together with a printout from routine RECUPJ of the lower triangle of transformed continuum-continuum Hamiltonian matrix blocks;
= 5 as for 4, together with a printout from routine RECUPJ of the transformed long-range potential coefficients.
IBUG6
= 0 normally;
= 1 for a printout of the *LS*-coupled bound-bound Hamiltonian matrix elements from STG2;
= 2 as for 1, together with a printout of the *LS*-coupled bound-continuum Hamiltonian matrix blocks from STG2;
= 3 as for 2, together with a printout of the *LS*-coupled continuum-continuum matrix blocks from STG2;
= 4 as for 3, together with a printout of the *LS*-coupled long-range potential coefficients from STG2.
IBUG7
= 0 normally;
= 1 for a printout from LSCONT of the partially transformed bound-bound Hamiltonian matrix elements;
= 2 as for 1, together with a printout from routine LSCONT of the partially transformed bound-continuum Hamiltonian matrix blocks;
= 3 as for 2, together with a printout from routine LSCONT of the lower triangle of partially transformed continuum-continuum Hamiltonian matrix blocks;
= 4 as for 3, together with a printout from routine LSCONT of the partially transformed long-range potential coefficients.
IBUG8
= 0 normally;
= 1 for a printout from routine SPINOR of the transformed bound-bound Hamiltonian matrix elements which now include the spin-orbit interaction;
= 2 as for 1, together with a printout from routine SPINOR of the transformed bound-continuum Hamiltonian matrix blocks which now include the spin-orbit interaction;
= 3 as for 2, together with a printout from routine SPINOR of the lower triangle of transformed continuum-continuum Hamiltonian matrix blocks which now include the spin-orbit interaction.
IBUG9
= 0 normally;
= 1 for a printout from routine SPINBB of the bound-bound Hamiltonian matrix elements of the spin-orbit operator in *LSJ*-coupling;
= 2 as for 1, together with a printout from routines SPINCB and SPINBC of the continuum-bound and

bound-continuum Hamiltonian matrix blocks of the spin-orbit operator in *LSJ*-coupling;
= 3 as for 2, together with a printout from routine SPINCC of the lower triangle of continuum-continuum Hamiltonian matrix blocks of the spin-orbit operator in *LSJ*-coupling.

4.6. Option: output of term-coupling coefficients on IPUNCH

Formatted output from routine BOUNDJ for JAJOM (Saraph [69,70]). It is assumed that the main production run to obtain reactance matrices is in *LS*-coupling, though the mass-correction and Darwin terms `JRELOP(1)` and `JRELOP(2)` can be either off or on as required. The term-coupling coefficient option described here must be considered a special one-off run terminating in RECUPD. The user must set the following parameters in the input data of:

STG1 set `JRELOP(1)` = `JRELOP(2)` = `JRELOP(3)` = 1;

STG2 set `JRELOP(1)` and `JRELOP(2)` as in the *LS*-coupling run, and set `JRELOP(3)` = -1; set also `NDIAG` = 1 and `NAST` = `NCFG`;

RECUPD set `IPUNCH` > 0.

Summary of the data records (the variable names are described in the glossary in Section 9):

Target terms:

1. `NAST`; format(I4);
2. `(I, ISAT(I), LAT(I), LPTY(I), I=1,NAST)`; format(I4,3I3).

Term-coupling coefficients for each target J_i (`JJ` $= 2 * J_i$):

3. `JJ, MTC, TEXT, NZ, NELC`; format(I10,I5,A32,I3,5X,I3);
4. `(ITMP(K), TEMP(K), K=1,MTC)`; format(5(I5,F9.6));

repeat records 3–4 for each `JJ` $= 2 * J$.

Terminator:

5. `0, 0`; format(I10,I5,5X,'TCC END').

Warning : the term coupling coefficient labels (`ITMP` in the output) may need to be modified before use in JAJOM.

4.7. Output of Hamiltonian matrices and coefficients on ITAPE3

The same format and variables are used as in the output on `ITAPE3` from module STG2 (see Section 3.5), except that the optional data (records 7 and 10) is no longer present.

`ISAT` = 0 and `NSPN` = 0 in the output are used to denote that the data are in intermediate-coupling.

4.8. Output of dipole matrices on ITAPE4

Output from routine DMOUT. The same format and variables are used as in the output on `ITAPE4` from module STG2 (see Section 3.6).

5. Module STGH

This module of RMATRX1 deals with the remaining 'inner-region' tasks. It can diagonalize the Hamiltonian matrices and process the dipole matrices from files created in *LS*- or in intermediate-coupling. In the latter case, when STG2 has been run with spin-orbit terms on specifying `JRELOP(3)` $\neq 0$, the files from STG2 *must* have been reprocessed by RECUPD. These files in the 'continuum function' representation of Eq. (14) and Eq. (48) are transformed to a form in which the $(N+1)$-electron Hamiltonian is diagonal.

Fig. 16 displays a flow diagram for the routines in STGH.

```
MNSTGH AASTGH DMAT  DMAT1 DA2
                          RECOV2
                          VMUL
                    DMAT2 DA2
              RSCT  DA2
                    MDIAG TASK1
                          TASK2
                          TASK3
                          TASK4
                    ORDER
              STG3RD
              TAPERD ORDER
                    RECOV2
              WRITOP ORDER
```

Fig. 16. Calling tree for module STGH. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

There are three main computational sections in STGH (the controlling routines are named in brackets):

- initialisation and reading input files (BLOCK DATA, STG3RD, TAPERD);
- diagonalizing the Hamiltonian in the continuum basis (RSCT);
- calculating electric transition data in the new representation (DMAT).

5.1. Routines

MNSTGH
is the program routine and contains all COMMON blocks used in STGH. It sets /MEMORY/ pointers, and calls AASTGH.

AASTGH
is called by MNSTGH and controls the STGH computation, invoking the three main computational sections as summarised above.

There is a loop over the $(N+1)$-electron system symmetries to diagonalize the Hamiltonian matrices, which is completed before entering the dipole matrix routine DMAT.

DA2
is used to write to the scratch file IDISC2 if there is insufficient memory in STGH. See the description of this routine in Section 2.

DMAT
called if IPOLPH > 1, is the controlling routine for photoionization, transforming the reduced dipole matrices $\boldsymbol{D}$ of Eq. (48) to a form in which the $(N+1)$-electron Hamiltonian is diagonal, Eq. (49):

$$\boldsymbol{M}(k,k') = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}_{k'}$$

- The $\boldsymbol{D}$ matrix is from module STG2 or RECUPD on file ITAPE1, in length and velocity forms (DEL and DEV). The same transformation is applied to the Buttle corrections to the dipole matrix.
- The $\boldsymbol{V}$ eigenvectors are from routine RSCT, in memory or scratch file IDISC2 via a call to DA2.
- The $\boldsymbol{M}$ matrix is output to ITAPE4, in length and velocity forms (DML and DMV), together with the transformed Buttle corrections.

Routine DMAT1 or DMAT2 is called to read the appropriate matrix elements from ITAPE1, to get the eigenvectors, to do the matrix multiplications, and to write the results to ITAPE4.

DMAT1
is called by DMAT if there is insufficient memory (`&MEM`), in which case the matrix multiplications are done using scratch file `IDISC3` to buffer intermediate matrices.

DMAT2
is called by DMAT if there is sufficient memory (`&MEM`), in which case the matrix multiplications are done in memory, in an optimum way.

MDIAG
is the same diagonalization routine as HSLDR, described in the STGLIB module. However, micro tasking instructions have been inserted (which appear as comments), in order to facilitate optimisation. It calls routines TASK1...4 to perform independent parallel tasks.

ORDER
returns a pointer array (`NORDER`) to define an ascending or descending set from an input array of real numbers (`EN`): i.e. let (`EN`(i), i = 1,`NDIM`) be in arbitrary order, then (`EN(NORDER`(i)), i = 1,`NDIM`) is in ascending or descending order.

It is used to order target energies and channels.

RECOV2
See the description of this routine in Section 2.

RSCT
controls the diagonalization of a Hamiltonian matrix.

The matrix is normally held in the array `HNP1` in the common block /MATRIX/. However, a rarely-used facility exists to read it from scratch file `IDISC1` in order to reduce its size in accordance with the user-specified parameter `NOT1`.

On calling the diagonalization routine MDIAG, the eigenvalues `VALUE`(k) $\equiv E_k$ of Eq. (13) are found. The surface amplitudes `WMAT`(i,k) $\equiv w_{ik}$ are obtained from the eigenvectors c_{ijk} as in Eq. (20)

$$w_{ik} = \sum_j c_{ijk} u_{ij}(\mathtt{RA})$$

where the u_{ij} boundary amplitudes, which were evaluated in module STG1, are transferred here via the input file `ITAPE2`.

If `IPOLPH` > 1, then the eigenvectors are required in routine DMAT in order to transform the dipole matrices. In this case the eigenvectors are either stored in memory if sufficient (`&MEM`), or on the scratch file `IDISC2` via a call to DA2.

STG3RD
reads the input data specifying the case. The input data is summarised in Section 5.3.

TAPERD
reads input data from the Hamiltonian matrix file from either STG2 or RECUPD. There is a facility here for adjusting the diagonal elements of the Hamiltonian matrix, before diagonalization.

TASK1...4
are primitive algorithms for diagonalization. Independent parallel tasks called by MDIAG:
TASK1 evaluates Eq. (125) – outer loop parallel, inner loops vectorised;
TASK2 evaluates Eq. (126) – outer loop parallel, inner loop vectorised;
TASK3 finds eigenvalues – each one independently;
TASK4 finds batches of eigenvectors – each one independently.

VMUL
is a matrix multiply routine.

WRITOP
writes basic data at the start of the H file ITAPE3.

5.2. Data files

The following is a summary of the data files required by STGH. The unit numbers and file names are defined in the program. Although the variables are part of the input data (for consistency with earlier versions of RMATRX) you need only supply dummy values i.e. set them to 0.

IREAD = 5 — 'STGH.INP'
File type: formatted sequential input.
Written by user.
Read by routine STG3RD.
Description: user supplied input, see Section 5.3 for details.

IWRITE = 6 — 'STGH.OUT'
File type: formatted sequential output.
Written throughout STGH.
Read by user.
Description: line-printer or job output — the log file.

IPUNCH = 0 NOT USED

IDISC1 = 11 (not normally used, i.e. only if NOT1 < NRANG2)
File type: sequential binary scratch file.
Written by routine TAPERD.
Read by routines RSCT.
Description: temporary store of Hamiltonian matrix.

IDISC2 = 12 (used only if IPOLPH = 2 and insufficient memory)
File type: direct access scratch file, record length 512 (8 byte) words.
Written by routine RSCT.
Read by routines DMAT1 and DMAT2.
Description: temporary store of eigenvectors.

IDISC3 = 13 (used only if IPOLPH = 2 and insufficient memory)
File type: direct access scratch file.
Written by routine DMAT1.
Read by routine DMAT1.
Description: internally used by DMAT1 to store dipole matrix data.

ITAPE1 = 1 (used if IPOLPH = 2) — 'STG2D.DAT' or 'RECUPD.DAT'
File type: binary sequential input.
Written by module STG2 or RECUPD.
Read by routines DMAT, DMAT1 and DMAT2.
Description: dipole matrix file.

ITAPE2 = 2 — 'STG2H.DAT' or 'RECUPH.DAT
File type: binary sequential input.
Written by module STG2 or RECUPD.
Read by routine TAPERD.
Description: Hamiltonian matrix file.

ITAPE3 = 3 — the H file, 'H.DAT'
File type: binary sequential output.
Written by routines WRITAP and RSCT.
Read by module STG4 and Seaton's codes STGB and STGF of the Opacity Project (Berrington et al. [11]).
Description: diagonalized Hamiltonian matrix data. See Section 5.6.

ITAPE4 = 4 — the D files, 'D00.DAT', 'D01.DAT', 'D02.DAT' ...(used if `IPOLPH` = 2)
File type: binary sequential output.
Written by routine DMAT.
Read by module STG4 and Seaton's codes STGBB and PREBF/STGBF of the Opacity Project (Berrington et al. [11]).
Description: dipole matrix files D*nn*, where D00 is a header file, followed by one D*nn* for each pair of initial and final $(N+1)$-electron symmetry that gives rise to electric dipole transitions. See Section 5.7.

STGH attempts to open the files RECUPH.DAT and RECUPD.DAT first. If these are not found then it attempts to open STG2H.DAT and STG2D.DAT.

5.3. Input data on IREAD

The user-supplied input data is read in routine STG3RD on input unit number `IREAD`. Free format is used, with one exception:

- FORMAT(18A4) for reading text into array `TITLE`.

Summary of the data records (the variable names are described in the glossary in Section 9):

1. `(TITLE(K),K=1,18).`
   I/O units, described in Section 5.2:
2. `IWRITE, IPUNCH, IDISC1, IDISC2, IDISC3,`
   `ITAPE1, ITAPE2, ITAPE3, ITAPE4.`
   Debug parameters, described in Section 5.5:
3. `NBUG1, NBUG2, NBUG3, NBUG4, NBUG5, NBUG6, NBUG7, NBUG8, NBUG9.`
   Controlling information:
4. `ICOPY, ITOTAL, IPOLPH;`
5. `NBUT, NOT1, NOT2, IDIAG, NEST, INAST.`
   Target energy adjustments, described in Section 5.4:
6. `EST(N),N=1,NEST` (only if `NEST` > 0).
   $(N+1)$-electron symmetry specification, see note below:
7. `LRGL, NSPN, NPTY` (only if `INAST` > 0).
   repeat record 7 for `INAST` > 0 symmetries.

Important note

- Inexperienced users should set `NBUT` = 1, `NOT1` = `NOT2` = `NRANG2`, `IDIAG` = 1 and `INAST` = 0. Setting `INAST=0` forces the program to loop over all available symmetries (normal operation).
- The user should not set `INAST` > 0 if `IPOLPH` = 2, as the code cannot skip over embedded pairs of transition matrices. However one may still select symmetries for electron collision calculations by specifying `INAST` > 0, followed by `INAST` values of `LRGL,NSPN,NPTY`.

5.4. Option: adjusting target energies

There is a facility in STGH for adjusting the target energies. A corresponding change is made to the diagonal elements of the $(N+1)$-electron Hamiltonian matrix before diagonalization. The adjusted energies are carried over to the external region, so that the asymptotic equations are solved with consistent channel energies.

This facility is normally used to allow observed energy splittings to be input to the scattering calculation — EST(I), I = 1, NEST are the observed target energies: in units of 2*Ry if EST(1) > 0, in Ry if EST(1) = 0.0, or in cm^{-1} wave numbers if the values EST are prefaced by a minus sign. The ordering of the target energies in this array must correspond exactly with the ordering of the target states defined by the user in modules STG2 or RECUPD (i.e. they may not necessarily be in order of increasing energy).

The justification for adjusting the Hamiltonian matrix can be seen by examining the $(N+1)$-electron Hamiltonian in Eq. (4). This clearly can be split into an N-electron part and terms involving the $(N+1)$th electron:

$$H^{N+1} = H^N - \tfrac{1}{2}\nabla^2_{N+1} - \frac{Z}{r_{N+1}} + \sum_{n=1}^{N} \frac{1}{r_{n,N+1}}$$

In order to diagonalize the $(N+1)$-electron Hamiltonian in Eq. (13), we have calculated in modules STG2 and RECUPD matrix elements involving the R-matrix expansion in Eq. (12). In particular, the continuum-continuum diagonal matrix element

$$\begin{aligned}\text{HNP1}(j,j)_{\text{CC}} &= (\mathcal{A}\overline{\Phi}_i \tfrac{1}{r} u_{ij} \mid H^{N+1} \mid \mathcal{A}\overline{\Phi}_i \tfrac{1}{r} u_{ij}) \\ &= (\Phi_i \mid H^N \mid \Phi_i) + h(N+1) \end{aligned} \tag{116}$$

i.e. separates into a sum of the target energy $E_i^N = (\Phi_i \mid H^N \mid \Phi_i)$ from Eq. (6) and a term involving the continuum orbital $h(N+1)$. Thus, accurate experimental values of the target energies E_i^N can be incorporated into the calculation by adjusting the diagonal elements of the $(N+1)$-electron Hamiltonian.

There are three practical points to bear in mind —

- The first target state defined in modules STG2 or RECUPD must be the one with the lowest energy, the ground state. The remaining states can be defined in arbitrary order, and STGH will reorder the states in ascending energy on output to the H and D files.
- If adjusting target energies, it is probably not valid to change the original energy order of the states. For example, if one state lies above another according to the calculated energies, but the states are actually observed to be the other way round, it is probably better to set the two energies degenerate to some average value.
- although continuum-continuum diagonal elements can be changed, there is no change to the diagonal elements from the bound-bound part of the matrix, which is normally associated with low lying resonances or bound states. This is somewhat inconsistent. It is recommended therefore to keep energy adjustments small.

5.5. Debug prints

Debugging prints are under the control of the NBUG parameters, specified in record 3 of the input data. Set these to zero in production runs, otherwise a large amount of output can be printed.

NBUG1 = 0 NOT USED.
NBUG2 = 0 NOT USED.
NBUG3 = 0 NOT USED.
NBUG4 = 0 NOT USED.
NBUG5 > 0 will provide a debug print from routine RSCT. The largest eigenvector components are printed for the NBUG5 lowest eigenvalues. This variable is also affected by NBUG8 so that NBUG5 = max(NBUG5, 1) when NBUG8 = 1; NBUG5 = MNP2 when NBUG8 = 2; and NBUG5 = 0 when NBUG8 = 3.
NBUG6
= 1 for a full printout of the $(N+1)$-electron Hamiltonian when read by TAPERD;
= 2 for a similar printout by RSCT on processing the Hamiltonian.

NBUG7
> 0 gives a print of the arrays AC, BLC and BVC from routine DMAT;
≥ 1 prints the length dipole matrix elements in routines DMAT1 and DMAT2;
≥ 2 prints the velocity dipole matrix elements in routines DMAT1 and DMAT2.

NBUG8
$= 2$ to print surface amplitudes;
$= 3$ to print all eigenvalues and eigenvectors of the $(N+1)$-electron Hamiltonian;
see also the description of NBUG5.
NBUG9 = 0 NOT USED.

If any of NBUG5, NBUG6 or NBUG8 are non-zero then the eigenvalues are printed relative to the ground-state in routine RSCT.

5.6. *Output on ITAPE3 — the H file*

Summary of the output records (the variable names are described in the glossary in Section 9):
Basic information from routine WRITOP:
1. `NELC, NZ, LRANG2, LAMAX, NAST, RA, BSTO;`
2. `(ENAT(N), N=1,NAST);`
3. `(LAT(N), N=1,NAST);`
4. `(ISAT(N), N=1,NAST);`
5. `((COEFF(K,L), K=1,3), L=1,LRANG2);`

Partial wave dependent data from routines WRITOP and RSCT:

6. `LRGL, NSPN, NPTY, NCHAN, MNP2, MORE;`
7. `(NCONAT(N), N=1,NAST);`
8. `(L2P(I),I=1,NCHAN);`
9. `(((CF(I,J,K), I=1,NCHAN), J=1,NCHAN), K=1,LAMAX);`
10. `(VALUE(K), K=1,MNP2);`
11. `((WMAT(I,K), I=1,NCHAN), K=1,MNP2);`

repeat records 6–11 for INAST $(N+1)$-electron symmetries.

5.7. *Output of dipole matrices on ITAPE4 — the Dnn files*

- Summary of the output records on D00.

An index of dipole allowed transitions from routine DMAT:
1. `KOUNT;`
2. `ISPN,ILRGL,IPTY, JSPN,JLRGL,JPTY`

repeat record 2 for KOUNT dipole allowed transitions.

- Summary of the output records on D*nn*.

A new file is opened for each dipole allowed transition, i.e. D01, D02, etc. .
1. `NOTERM,MNP2,NCHAN,JLRGL,JPTY,JSPN,MMNP2,MCHAN,ILRGL`
2. Dipole matrix elements from routines DMAT1 and DMAT2:
   a. set `N2=0` initially;
   b. set `N1 = N2+1` and `N2 = min(N2+NOTERM,MMNP2)`, and `M2=0`;
   c. set `M1 = M2+1` and `M2 = min(M2+NOTERM,MNP2)`;
   d. `((DML(M,N), M=M1,M2), N=N1,N2), ((DMV(M,N), M=M1,M2), N=N1,N2);`

repeat records c–d for `JK = 1, 1+(MNP2−1)/NOTERM`;
repeat records b–d for `IK = 1, 1+(MMNP2−1)/NOTERM`.
Dipole matrix Buttle corrections from routine DMAT:

```
3. ((BBUTL(M,N), N=1,MCHAN), M=1,MNP2),
   ((BBUTV(M,N), N=1,MCHAN), M=1,MNP2);
4. ((ABUTL(M,N), N=1,NCHAN), M=1,MMNP2),
   ((ABUTV(M,N), N=1,NCHAN), M=1,MMNP2);
5. ((BBUTL(M,N), N=1,NCHAN), M=1,MCHAN),
   ((BBUTV(M,N), N=1,NCHAN), M=1,MCHAN).
```

Clebsch-Gordan coefficients and outer region contributions, copied from file ITAPE1:

```
6. MAXM1,(CGC(M),M=1,MAXM1);
7. ((AC(M,N), N=1,MCHAN), M=1,NCHAN),
   ((BLC(M,N), N=1,MCHAN), M=1,NCHAN),
   ((BVC(M,N), N=1,MCHAN), M=1,NCHAN).
```

6. Module STG4

This module of RMATRX1 deals with the 'external-region' tasks. It solves the external region coupled equations (see Eq. (30)), for either atomic or ionic targets, and matches to the R-matrix on the internal region boundary to produce collision observables.

This version replaces those portions of the former STG3 [12,13,73] that dealt with tasks outside the R-matrix boundary RA. We prefer the philosophy used in the Opacity Project [11], where the 'outer region' tasks were separated and performed in a new suite consisting of the modules STGB, STGBB, STGF, and STGBF (letter B refers to solutions with bound state asymptotic boundary conditions while letter F refers to a boundary condition representing a free electron in the field of an N-electron target). However, these Opacity Project outer region programs are at present unpublished; moreover they are restricted to ionic targets. We therefore include here a less efficient package, based on the published coupled differential equation solver of Crees [37], in order that RMATRX1 can yield a range of output, for both ionic and neutral targets:

- electron collision strengths (output in file XOMEGA);
- photoionization cross sections from ground states (file XSECTN);
- frequency dependent polarizabilities (file XBOUND);
- bound state energies and f-values (file XBOUND);
- other optional output, such as partial cross sections, T-matrices , K-matrices , eigenphases, photoelectron asymmetry parameter β, etc. (file XDUMP).

STG4 cannot however calculate with excited bound states; further, no external region contribution to the dipole matrices as in Eq. (52) can be applied.

Fig. 17–18 displays a flow diagram for the routines in STG4.

STG4 must be linked with STGLIB and the asymptotic package, CREES, in order to form an executable program. Routines AFACE, CG, DRACAH, FACTT and HSLDR plus any routines called by them are in STGLIB and CREES.

There are three main computational sections in STG4 (the controlling routines are named in brackets):

- initialisation and reading input files (STG4RD, READH1, READD1, MESHE, READH2);
- bound state options and further initialisation (BOUNDE, OUT1, READD2);
- calculating collisional data for each impact energy (ENERGY).

```
 1 MNSTG4 AASTG4 BOUNDE ITERE  AFACE(*) ----
 2                            MA01A
 3                            RMAT   BUTO
 4                                   MA01A
 5                            VMUL
 6                     WVFNIN
 7              ENERGY AFACE(*) ----
 8                     BETA   CG(*)
 9                            COULGA
10                            DRACAH(*)
11                            FACTT(*)
12                            SHRIEK
13                     EXTRAP
14                     KMAT   MA01A
15                            VMUL
16                     OUT2
17                     PHASE4 HSLDR(*)
18                     PHOT   MA01A
19                            VMUL
20                     POLZ
21                     RMAT   > 3
22                     XOMEGA MA01A
23                            VMUL
24              MESHE
25              OUT1   OUTJJ
26              READD1
27              READD2 DMUL
28                     READD3
29              READH1
30              READH2 READH3
31              STG4RD
```

Fig. 17. Calling tree for module STG4. The routines are given in alphabetical order within each branch **not** the order in which they are actually called. The routines marked with (*) are in module STGLIB or CREES.

```
 1 AFACE  ASYPCK ASYMPT ASYSM  COULGM
 2                            EXPAN
 3                            ITERA  POP    BOOLE
 4                                          DCHAIN
 5                                   PPFS
 6                                   QROP   DCHAIN
 7                                   ZETA
 8                            PHAS
 9                            POTS
10                            SOLV   WOUTER
11                     FG     MTINV
12                            WW
13                     SOLV   > 10
14              BOUNDC
15              INTERP
16              OMEGA
17              SOLNS  PRTFNS
18                     SECDRV WW
```

Fig. 18. Calling tree for module STG4, CREES section.

6.1. Routines

MNSTG4
is the program routine and contains all common blocks used in STG4. It sets /MEMORY/ pointers and calls AASTG4.

AASTG4
is called by MNSTG4 and controls the STG4 computation, invoking the three main computational sections as summarised above.

BETA
is only called if IRAD≥1, for optional calculation of photoionization cross sections to each final ionic state and/or the photoelectron asymmetry parameter β.

The cross section to a final state j is given by

$$\sigma_j = \left(\tfrac{4}{3}\pi^2 a_0^2 \alpha\right) \left(\frac{\omega C}{2L_0+1}\right) \sum_{l_j L} |\left(\Psi_j^- \,||\, D \,||\, \Psi_0\right)|^2$$

where $C = 1$ in the length form, and $C = 4/\omega^2$ in the velocity form, with the photon energy (ω) being in Ry. $\boldsymbol{D}$ is a general dipole operator which could be either the length or velocity operators of Eq. (43); the dipole matrix associated with the initial state Ψ_0 is input to this routine in the DSTORE array, being calculated in routine PHOT.

The asymmetry parameter is defined as the $l = 2$ term in the expression for the differential cross section for photoionization when the incident radiation is linearly polarized:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\hat{k}} = \frac{\sigma}{4\pi}[1 + \beta P_2(\cos\theta)]$$

where θ is the angle of the ejected electron relative to the axis of polarization. β is determined from (Burke and Robb [22]):

$$\begin{aligned}\beta = \; & \left(4\pi^2 a_0^2 \alpha\right) \left(\tfrac{C\omega}{\sigma(2L_0+1)}\right) (-1)^{L_f+L_0} \\ & \times \sum_{l_f l_{f'} LL'} \mathrm{i}^{l_f - l_{f'}}\, \mathrm{e}^{-\mathrm{i}\sigma_{l_f} + \mathrm{i}\sigma_{l_{f'}}} \\ & \times [(2l_f+1)(2l_{f'}+1)(2L+1)(2l'+1)]^{1/2} \\ & \times W(Ll_fL'l_{f'};L_f2)\; W(1L1L';L_02)\; C(l_fl'_f2;00)\; C(112;00) \\ & \times (\Psi_0 \,||\, D \,||\, \Psi_{j'}^-)\, (\Psi_j^- \,||\, D \,||\, \Psi_0)\end{aligned} \qquad (117)$$

BOUNDE
is the controlling routine for calculating bound state energies E_0 and wavefunction expansion coefficients A_{0k} in Eq. (50) for photoionization etc. . For each required symmetry, the lowest bound state is calculated using an iterative procedure in routine ITERE. Routine WVFNIN is then called to calculate the wavefunction expansion coefficients.

BUTO
is a real function used to calculate the Buttle correction for a given channel energy, using the procedure of Seaton [77].

COULGA
is called by BETA to calculate the Coulomb phase shift $\sigma_l = \arg\Gamma(l+1+i\eta)$ where $\eta = -z/k$.

DMUL
is called by READD2 to calculate a gf-value as in Eq. (55) for a dipole allowed transition by multiplying the dipole vector associated with the initial state, calculated in routine READD3, by the expansion coefficients (A_{fk}) defining a final state vector, calculated in BOUNDE:

$$gf = g\left(\frac{E_0-E_f}{3}\right) C \left| \sum_k A_{fk} (\psi_k \, || \, D \, || \, \Psi_0) \right|^2 \tag{118}$$

where $C = 1$ for the length form, and $C = 4/(E_0 - E_f)^2$ in the velocity form; though if $E_0 \approx E_f$ the velocity form is set zero. The energies of the initial and final states, E_0 and E_f, are in Ry, and $g = (2L+1)(2S+1)$ or $g = (2J+1)$ depending on the coupling scheme.

Results can be inaccurate because an external region contribution to the dipole matrix elements is not calculated. The effect of this can be reduced by enlarging the internal region.

ENERGY
is the controlling routine for solving the external region equations to obtain the K-matrix from the R-matrix , and hence the cross sections etc. .

A single scattering energy is treated in the routine. The partial wave loop is then entered in order to obtain:
• (optional) frequency dependent polarizabilities (routine POLZ).
• the R-matrix (routine RMAT);
• external region solutions (routine AFACE);
• the K-matrix (routine KMAT);
• collision strength (routine XOMEGA);
• photoionization cross sections (routine PHOT);
• (optional) output of K-matrices and other data on file XDUMP (routine OUT2).

After the loop over partial waves, the collision strengths can be 'topped-up' to output the total collision strengths (routine EXTRAP). In the case of photoionization, the cross section for each final state and the β asymmetry parameter can optionally be output (routine BETA).

EXTRAP
estimates the collision strength contribution from a given total angular momentum (LTOP) to infinity, and returns the total 'topped-up' collision strength. The partial collision strength at LTOP and at LTOP−1, and the partial sum to LTOP, are required as input.

Each transition is considered in turn. If the transition is spin-allowed, or a fine-structure one, and the initial and final states are non-degenerate, the collision strength is assumed to form a geometric series (cf. Burgess et al. [19]): $\Omega_J \sim x^J$, and the sum from LTOP to infinity is $\Omega_{\mathrm{LTOP}}(x/(1-x))$. Here $x = \Omega_{\mathrm{LTOP}}/\Omega_{\mathrm{LTOP}-1}$, unless this yields a value greater than one or the transition is dipole allowed, in which case x is taken as the ratio of the final and initial electron energies. Transitions involving degenerate states are examined separately.

ITERE
is called by BOUNDE to determine the energy of a bound state, e.g. the initial state in a photoionization calculation.

Since all the channels are closed the required solution is given by Eq. (40):

$$F_i = \sum_{j=1}^{\mathrm{NCHAN}} c_{ij} x_j, \qquad i = 1, \mathrm{NCHAN}, \qquad r > \mathrm{RA}$$

where the c_{ij} are NCHAN independent solutions of the asymptotic equations with exponentially decaying boundary conditions. The coefficients x_j are determined by requiring that the solution F_i satisfies the R-matrix boundary condition, as in Eq. (41)

$$F_i(\mathtt{RA}) = \sum_{j=1}^{\mathtt{NCHAN}} R_{ij}\left(r\frac{\mathrm{d}F_j}{\mathrm{d}r} - \mathtt{BSTO}\cdot F_j\right)_{r=\mathtt{RA}}, \qquad i = 1,\mathtt{NCHAN}$$

This gives a homogeneous equation for the x_j

$$\sum_{j=1}^{\mathtt{NCHAN}} B_{ij}x_j = 0, \qquad i = 1,\mathtt{NCHAN}$$

where the B-matrix is defined by Eq. (42)

$$B_{ij} = c_{ij}(\mathtt{RA}) - \sum_{j'}^{\mathtt{NCHAN}} R_{ij'}\left(r\frac{\mathrm{d}c_{j'j}}{\mathrm{d}r} - \mathtt{BSTO}\cdot c_{j'j}\right)_{r=\mathtt{RA}}$$

This equation only has non-trivial solutions at the eigenenergies corresponding to the bound states of the system. To find these energies we set $x_1 = 1$, solve the remaining NCHAN-1 equations

$$\sum_{j=2}^{\mathtt{NCHAN}} B_{ij}x_j = -B_{i1}, \qquad i = 2,\mathtt{NCHAN}$$

and look for solutions of the non-linear equation

$$f(E) = \sum_{j=1}^{\mathtt{NCHAN}} B_{1j}x_j = 0$$

The procedure adopted is to first find by a stepping procedure two energies which bracket the required zero in $f(E)$. A bisection process is then started with these two energies as end-points and in this way the zero of $f(E)$ is located between successively narrower limits. The bisection process is continued until $f(E)$ has a magnitude less than some predetermined value (e.g. 10^{-3}). The program then switches to Newton's iterative method until $f(E)$ is less than a second predetermined value (e.g. 10^{-8}).

The lowest R-matrix pole is usually closely associated with the ground state eigenenergy of the system, so we start with a trial energy close to this pole. We use a procedure developed by Burke and Seaton (1984) and Seaton (1985) for carrying out the calculation in the vicinity of the R-matrix pole.

The final value of the bound state energy obtained is returned (EI), as is an array (D) containing information on the bound state solution on the boundary: $\left(r\mathrm{d}F_j/\mathrm{d}r - \mathtt{BSTO}\cdot F_j\right)_{r=\mathtt{RA}}$, needed for the calculation of the bound state wavefunction in routine WVFNIN.

KMAT
calculates the K-matrix from the R-matrix and asymptotic solutions on the boundary. Let:
s_{ij}, $j = 1,\mathtt{NOPEN}$ be the regular (sine) solutions, input to the routine in array F($i,j,1$);
c_{ij}, $j = 1,\mathtt{NOPEN}$ be the irregular (cosine) solutions, input in F($i,j,2$);
c_{ij}, $j = \mathtt{NOPEN}+1,\mathtt{NCHAN}$ be the exponentially decaying solutions, input in F($i,j,1$);
for $i = 1,\mathtt{NCHAN}$. The derivatives are input in corresponding places in the FD array. We then solve the simultaneous equations for the K-matrix as in Eq. (36):

$$\sum_{l=1}^{\mathtt{NCHAN}}\left[c_{il}(\mathtt{RA})-\sum_{m=1}^{\mathtt{NCHAN}}R_{im}\left(r\frac{\mathrm{d}c_{ml}}{\mathrm{d}r}-\mathtt{BSTO}\cdot c_{ml}(r)\right)_{r=\mathtt{RA}}\right]K_{lj}$$
$$=-\left[s_{ij}(\mathtt{RA})-\sum_{m=1}^{\mathtt{NCHAN}}R_{im}\left(r\frac{\mathrm{d}s_{mj}}{\mathrm{d}r}-\mathtt{BSTO}\cdot s_{mj}(r)\right)_{r=\mathtt{RA}}\right] \tag{119}$$

for i=1,NCHAN and j=1,NOPEN; i.e.

$$\boldsymbol{B}\cdot\boldsymbol{K}=\boldsymbol{A}$$

where the $\boldsymbol{B}$ and $\boldsymbol{A}$ matrices are defined by the quantities in the first and second square brackets respectively of Eq. (119). The multiplication of the matrices are carried out in routine VMUL, and the simultaneous equations solved in MA01A.

MA01A
is a linear equation solver. It is called by routines ITERE, KMAT, PHOT and XOMEGA.

MESHE
is taken from program STGF (Berrington et al. [14]), and generates an energy mesh for the incident electron, based on equal intervals in the effective n below each threshold.

OUT1
opens the output files and writes header information.

OUT2
is called by ENERGY to write partial-wave dependent data (e.g. K-matrices) to output file XDUMP, under control of the user-supplied parameter IPRINT.

OUTJJ
is called by OUT1 to write JAJOM data header information, containing target state data etc. (Saraph [69,70]), if requested by IPRINT.

PHASE4
determines the eigenphases by diagonalizing the K-matrix to produce $\tan\delta_i$, where δ_i is the eigenphase in channel i. It returns also the eigenphase sum over the open channels. Note that the eigenphase as computed is arbitrary in multiples of π.

PHOT
calculates the photoionization cross section for a given photoelectron energy k^2, which is related to the photon energy ω through the conservation of energy, as in Eq. (10):

$$\omega+E_0=k^2+\epsilon_1$$

where E_0 and ϵ_1 are the initial bound state energy and ground state energy of the final ion respectively, in Ry. The total photoionization cross section is (cf. Eqs.(56–57))

$$\sigma(\omega)=\left(\tfrac{4}{3}\pi^2a_0^2\alpha\right)\left(\tfrac{\omega C}{2L_0+1}\right)\sum_{jl_jL}|\left(\Psi_j^-\,||\,D\,||\,\Psi_0\right)|^2$$

where $C=1$ in the length form, and $C=4/\omega^2$ in the velocity form, with the photon energy (ω) being in Ry. $\boldsymbol{D}$ is a general dipole operator which could be in either the length or velocity operators of Eq. (43). Expanding Ψ_j^- in terms of R-matrix states as in Eq. (58), we find that

$$(\Psi_j^- \, || \, D \, || \, \Psi_0) = \frac{1}{\mathrm{RA}} \sum_{k=1}^{\mathrm{MNP2}} \frac{(\psi_k \, || \, D \, || \, \Psi_0)}{E_k - E_0 - \omega} \, w_k^T(\mathrm{RA}) \, R^{-1} \, y^-(\mathrm{RA}) \tag{120}$$

where $w_k(\mathrm{RA})$ are the surface amplitudes input on the H file, $(\psi_k \, || \, D \, || \, \Psi_0)$ are the reduced matrix elements calculated in routine READD3, and y_{ij}^- is constructed from the asymptotic solutions such that it satisfies the boundary conditions

$$F^- \underset{r \to \infty}{\sim} (\pi k)^{-1/2} (\sin\theta + \cos\theta K)(1 + \mathrm{i}K)^{-1}$$

corresponding to a Coulomb modified plane wave plus ingoing spherical wave. These solutions are just linear combinations of the solutions defined by Eq. (31).

The vector $| \, (\Psi_j^- \, || \, D \, || \, \Psi_0) \, |$, which has real and imaginary components, and length and velocity forms, is stored in DSTORE for later use in routine BETA.

POLZ
calculates the frequency dependent dipole polarizability in both the length and velocity forms from the reduced dipole matrix elements. It is only called if IRAD=3.

The contribution to the polarizability from a continuum state, with given total angular momentum and spin, is defined by (Allison et al. [2])

$$\alpha_{M_L}(\omega) = 4(\mathrm{CGC})^2 \sum_k^{\mathrm{MNP2}} \frac{(E_k - E_0) C_k \, | \, D_k \, |^2}{(E_k - E_0)^2 - \omega^2} \tag{121}$$

where $C_k = 1$ in the length form, and $C_k = 4/(E_k - E_0)^2$ in the velocity form. The coefficient $\mathrm{CGC} = C(L_a 1 L; M_L 0)/(2L+1)^{1/2}$ is defined in module STG2. L_a is the total orbital angular momentum of the atomic state and L of the continuum state. The range of M_L is from zero to the minimum of L_a and L. The D_k dipole matrix elements, in length and velocity form, are defined in routine READD3. Energies are in Ry.

READD1
reads the D00 file, containing an index to the initial and final symmetry on each file of dipole matrices.

READD2
is the controlling routine for reading all available D files associated with a specific initial state symmetry, i.e. for the dipole allowed transitions; looping over the final state symmetries which satisfy the selection rules, and calling READD3 to read the reduced dipole matrices from module STGH and to calculate dipole vectors associated with the initial state. There is an optional call also to DMUL to use these vectors in calculating gf-values.

Two calls to READD3 are made: the first call opens the appropriate D file and reads the first record to obtain the sizes of the matrices, required for memory management; the second call reads the remainder of the D file and returns the dipole vectors, which are stored in memory for later use in routine PHOT.

READD3
is called by READD2 to read a specific D file for the reduced dipole matrix $(\psi_k \, || \, D \, || \, \psi_{k'})$ and multiplies by the initial state wavefunction coefficients A_{0k} from routine WVFNIN to return the dipole vector associated with a given initial state:

$$(\psi_k \, || \, D \, || \, \Psi_0) = \sum_{k'=1}^{\mathrm{MNP2D1}} (\psi_k \, || \, D \, || \, \psi_{k'}) A_{0k'}$$

for k=1,MNP2D2, where MNP2D1 and MNP2D2 are the sizes of the R-matrix bases in the initial and final states respectively. This dipole vector, in its length and velocity forms, is used in bound-bound and bound-free calculations in routines DMUL and PHOT.

READH1
reads the header of the H file, for the basic data, target states etc.

READH2
is the controlling routine for reading the whole of the remainder of the H file into memory. Routine READH3 is called for each partial wave to read the data; the long-range potential coefficients ($a_{ij}^{(\lambda)}$ in Eq. (29)), R-matrix poles and surface amplitudes (E_k and w_{ik} in Eq. (26)), are stored in memory for those partial waves required for the run.

Two calls to READH3 are made: the first call reads a single record to specify the partial wave symmetry and the sizes of the matrices, required for memory management; the second call reads the remainder of the data for this partial wave.

READH3
is called by READH2 to read the H file for a specific partial wave.

RMAT
calculates the Buttle-corrected R-matrix on the boundary for a given energy E in Ry, using Eq. (26), from the surface amplitudes w_{ik} (array WMAT) and eigenvalues E_k (array VALUE) of the Hamiltonian matrix. The correction R_{ii}^c to the diagonal elements of the R-matrix for each channel i whose energy is k_i^2 is calculated from an analytic fit in k_i^2 as described by Seaton (1987). The routine also returns these channel energies and the number of energetically open channels.

There are two further options available:

- a pole can be omitted in the evaluation of the R-matrix , and the amplitude ($w_{in}w_{jn}/\mathrm{RA}$) returned, where n is the omitted pole – this facility is used in routine ITERE to find a bound state lying close to an R-matrix pole;
- the inverse of the R-matrix (R^{-1}) can be returned – this is used in the photoionization calculation in routine PHOT.

SHRIEK
evaluates and stores factorials, GAMMA($i+1$) = $i!$, stored in common block /FACT/, for use in evaluating Clebsch-Gordan coefficients.

STG4RD
reads all the user-supplied input data from unit 5. See Section 6.3 for a description of the data.

VMUL
performs a matrix multiplication $\boldsymbol{C} = \boldsymbol{A}.\boldsymbol{B}$.

WVFNIN
is called by BOUNDE to determine normalised expansion coefficients A_{0k} for the bound state wavefunction in Eq. (50) (retaining for completeness the logarithmic derivative b), where the coefficients A_{0k} corresponding to a bound state energy of E_0 are defined by

$$A_{0k} = \frac{1}{\mathrm{RA}(E_k - E_0)} \sum_{j=1}^{\mathrm{NCHAN}} w_{jk} \left[r\frac{\mathrm{d}F_j}{\mathrm{d}r} - \mathrm{BSTO}\,F_j \right]_{r=\mathrm{RA}}$$

The quantity in the square brackets is input to the routine, having being calculated in routine ITERE from the bound state external region solutions on the boundary,

XOMEGA
calculates the partial collision strength from the K-matrix for a given electron energy. Updates also the partial sum. First the T-matrix is calculated, see Eq. (37):

$$T = \frac{1 + iK}{1 - iK} - 1 = -\frac{2K^2}{1 + K^2} + i\frac{2K}{1 + K^2}$$

then the partial collision strength, see Eq. (38):

$$\Omega_{ij}(k_2) = \frac{g}{2}\sum_{l_i l_j} | T |^2$$

where $g = (2L + 1)(2S + 1)$ or $g = (2J + 1)$ depending on the coupling scheme.

6.1.1. AFACE

AFACE provides an interface between STG4 and the asymptotic package. It is called from routines ENERGY and ITERE, to set up a call for a given energy to the external region program. STG4 uses the Crees program [37], which solves the coupled differential equations (see Eq. (30)) for both neutral and ionic targets, to yield the asymptotic functions and derivatives on the boundary $r = \mathrm{RA}$. Any other suitable package could be used here.

6.2. Data files

The following is a summary of the data files required by STG4. The unit numbers and file names are defined in the program.

5 — the user input — 'STG4.INP'
File type: formatted sequential input.
Written by user.
Read by routine STG4RD.
Description: user-supplied options, described in Section 6.3.

6 — the log file — 'STG4.OUT'
File type: formatted sequential output.
Written throughout STG4.
Description: line-printer or job output.

10 — the H file, 'H.DAT'
File type: binary sequential input.
Written by module STGH.
Read by routines READH1 and READH3.
Description: the diagonalized Hamiltonian matrices (see Section 5.6).

99 — the D files, 'D00.DAT', 'D01.DAT', 'D02.DAT' ...
File type: binary sequential input.
Written by module STGH.
Read by routines READD1 and READD3.
Description: D00 is an index file, the other D files each contain the reduced dipole matrix for a given pair of initial and final $(N + 1)$-electron symmetry that gives rise to electric dipole transitions (see Section 5.7).

1 — the file 'XOMEGA.OUT'
File type: formatted sequential output.
Written by routines OUT1 and ENERGY.
Description: total collision strengths for electron scattering as a function of energy (described in Section 6.5).

2 — the file 'XSECTN.OUT'
File type: formatted sequential output.
Written by routines OUT1 and ENERGY
Description: total cross sections for photoionization as a function of photon energy (described in Section 6.6).

3 — the file 'XBOUND.OUT'
File type: formatted sequential output.
Written by routines OUT1, DMUL and POLZ.
Description: bound state data, gf-values, and frequency dependent dipole polarizabilities (described in Section 6.7).

7 — the file 'XDUMP.OUT'
File type: formatted sequential output.
Written by routines OUT1, BETA, OUT2 and OUTJJ.
Description: partial-wave dependent data, such as K-matrices (described in Section 6.8).

6.3. Input data on unit 5

The user-supplied input data is read in routine STG4RD on input unit number 5. Free format is used.
Summary of the data records (the variable names are described in the glossary in Section 9):

Printout options, described in Section 6.4:
1. `IPRINT, IRAD, IPERT.`

   Accuracy parameters:
2. `AC;`
3. `RONE.`

   Electron energy mesh and options 5, 6 or 7:
4. `IMESH;`

   if `IMESH=1` or 0 then generate an equal energy interval mesh:
5. `MXE,EO,EINCR;`
6. if `IMESH=2` or <0 then generate an equal effective n energy mesh:
   a. `DQN;`
   b. `QNMAX;`
   c. `EMIN;`
   d. `EMAX;`
   e. `DEOPEN;`
   f. `IRDEC;`
7. if `IMESH=3` then read energy mesh:
   a. `MXE, QNMAX;`
   b. `EMESH`(k);
   repeat record b for k = 1, `MXE`.

   Partial wave options:
8. `IOPT1;`

   if `IOPT1` = 2 then read partial wave symmetries:
9. `IS, IL, IP;`

   repeat record 9 until end of file.

6.4. Printout options

`IPRINT`
> 0 for debug output on unit 6,

< 0 to create the XDUMP output file:
$= -1$ photoionization cross sections to each final state;
$= -2$ β asymmetry parameter for the photoelectron;
$= -3$ partial collision strengths;
$= -4$ eigenphases;
$= -5$ K-matrices (compact format);
$= -6$ T-matrices ;
$= -7$ K-matrices (JAJOM format);
$= -8$ JAJOM header file (no TCC);
$= -9$ JAJOM header file (TCC switch activated).

`IRAD`
$= 0$ for electron collisions (produces file XOMEGA);
$= 1$ for electron collisions and photoionization cross sections (files XOMEGA and XSECTN);
$= 2$ for photoionization (XSECTN);
$= 3$ for polarizabilities (XBOUND);
$= 4$ for bound state data (XBOUND).

`IPERT`
$= 0$ to switch off long-range coupling, > 0 to include coupling:
$= 1$ no top-up;
$= 2$ top-up, i.e. include estimate for higher partial waves in total collision strength.

6.5. *XOMEGA output file*

Summary of the output records for collision strengths (the variable names are described in the glossary in Section 9):
1. `NZED, NELC, NAST`
2. `ENAT(i), i=1,NAST`
3. `ETOT, NTRAN, (OMEGA(n), n=1,NTRAN)`
   repeat record 3 for all impact energies.

6.6. *XSECTN output file*

Summary of the output records for photoionization (the variable names are described in the glossary in Section 9):
1. `NZED, NELC, NAST`
2. `ENAT(i), i=1,NAST`
3. `IS1, IL1, IP1, I1`
4. `E1, NEN`
5. column headings for the following photoionization cross section records:
6. `EP, XL, XV`
   repeat record 6 for all photon energies.

6.7. *XBOUND output file*

Summary of the output records for bound state data (the variable names are described in the glossary in Section 9):
1. `IPRINT, IRAD, IPERT, NZED, NELC`
   If `IRAD` = 3 then output polarizabilities:

1a. column headings for the following polarizability records:
1b. `OMEGA2, M, ALPHL, ALPHV`
repeat this last record for each `M` and *L*.
If `IRAD` = 4 then output bound state energies and *gf*:
2. `AC`
3. `RONE`
4. `2`
5. `IS1, IL1, IP1, I1`
6. `1, 0.0`
7. `EI, EFFN`
repeat records 5-7 for all bound state symmetries considered, terminating with:
8. `0, 0, 0, 0`
9. `IS1, IL1, IP1, IS2, IL2, IP2, NN`
10. `I1, I1, EI, EF, FL, FV`
repeat records 9 and 10 for dipole allowed transitions.

6.8. XDUMP output file

The variable names are described in the glossary in Section 9.

6.8.1. Summary of output records for photoionization, IPRINT = −1
1. `NZED, NELC, NAST`
2. `ENAT(i), i=1,NAST`
3. `IS1, IL1, IP1, I1`
4. `E1, NEN`
5. column headings for the following photoionization cross section records:
6. `W1, (SIGL(i), i=1,NAST)`
repeat the last record for all photon energies `W1`.

6.8.2. Summary of output records for β parameters, IPRINT = −2
1. `NZED, NELC, NAST`
2. `ENAT(i), i=1,NAST`
3. `IS1, IL1, IP1, I1`
4. `E1, NEN`
5. column headings for the following β parameter records:
6. `W1, SIGL(N), BETALR, BETALI, N, SIGV(N), BETAVR, BETAVI`
repeat the last record for `N = 1, NAST` and for all photon energies `W1`.

6.8.3. Summary of output records for partial omegas, IPRINT = −3
1. `NELC, NZED, NAST, RA, LRANG2`
2. column headings for the following partial collision strength records:
3. `IS, IL, IP, E, ((XPART(i,j), i=1,j), j=1,IEND)`
repeat the last record for all partial waves `IS, IL, IP`, and energies `E`.

6.8.4. Summary of the output records for eigenphases, IPRINT = −4
1. `NELC, NZED, NAST, RA, LRANG2`
2. column headings for the following eigenphase records:

3. IS, IL, IP, E, PHSUM, (EIG(i), i=1,NOPEN)
   repeat the last record for all partial waves IS, IL, IP, and energies E.

6.8.5. Summary of the output records for K-matrices IPRINT = −5

1. NELC, NZED, NAST, RA, LRANG2
2. ITEMS, MPTY, IL, 0, E
3. ((RK(j,i), j=1,i), i=1,NOPEN)
   repeat the last two records for all partial waves IS, IL, IP, and energies E.

6.8.6. Summary of the output records for T-matrices , IPRINT = −6

1. MPTY, IL, ITARG(i), ITARG(j), TR(i,j), TI(i,j), 0, 4, E,' T'
   FORMAT(I2,I3,2I6,F16.8,3X,2X,F12.8,I3,I2,F12,8,A4)
   repeat this record for j = 1, i; i = 1, NOPEN;
   repeat for all partial waves IS, IL, IP, and energies E.

6.8.7. Summary of the output records for K-matrices (JAJOM format), IPRINT = −7

1. MPTY, IL, ITARG(i), ITARG(j), RK(i,j),' ',' ', 0.0, 0, 0, E,' K'
   FORMAT(I2,I3,2I6,E16.8,3X,2A1,F12.8,I3,I2,F12,8,A4)
   repeat this record for j = 1, i; i = 1, NOPEN;

   if IRAD≥1 then output dipole matrices:
2. MPTY, IL, 0, ITARG(i), DL(i,j), 'R', 'L', E1, IL1, 2, E,' D'
3. MPTY, IL, 0, ITARG(i), DV(i,j), 'R', 'V', E1, IL1, 2, E,' D'
   repeat records 2 and 3 for i = 1, NOPEN;
   repeat for all partial waves IS,IL,IP, and energies E.

6.8.8. Summary of the output records for JAJOM, IPRINT = −8 or −9

1. NAOP, IREFL, IW, IB, ITOP, MAT, IONQ, IPART
2. I, IS(I), LAT(I), ENAT(I), ISAT(I), LSPECT(LAT(I)), PARITY(LPTY(I))
   repeat this last record for I = 1, NAOP
3. NZED, NELC
4. LRANG2, ISLM, NOX
5. (I, I=0,LRANG2-1)
6. (IST(i), ILT(i), i=1,ISLM)
7. NZED, NELC
8. NEN, NEN
9. (K, EMESH(K), K=1,NEN)
10. NZED, NELC, 0
11. ' F.S. ', JJFSL, IFIT, JPUN
12. (FJ(j), j=1,JJFSL)
13. NTCC, NTCC (= 0 if IPRINT = −8)
14. MPTY, IL, ITARG(i), ITARG(j), RK(i,j),' ',' ', 0.0, 0, 0, E,' K'
    repeat this last record for j=1,i; i=1,NOPEN;
    and repeat for all partial waves IS,IL,IP, and energies E.

For a detailed description of the data output for IPRINT = −8 or −9 from routine OUTJJ, see the write-up of the program JAJOM (Saraph [69,70]).

7. Library STGLIB

This routine library is part of RMATRX1. Modules STG2, RECUPD and STG4 must be linked with STGLIB in order to form executable programs. No call is made from the library to any routine outside the library, apart from the usual FORTRAN functions.

Rather than describe the routines in alphabetical order they have been grouped together for convenience.

1. General
2. CFP — coefficients of fractional parentage
3. Angular integrals (FANO,HOWTS,TENSOR)
4. HSLDR — matrix diagonalization
5. NJGRAF — recoupling package

Only a brief description of the main routines in each group is given. These routines have all been documented before in the CPC program library.

7.1. GENERAL

BLOCK DATA — BDLIB1
defines the data in common blocks /TERMS/ and /CONSTS/. The data in /TERMS/ holds the allowed quantum numbers for arbitrary numbers of electrons in s, p and d shells, and for one electron in f shells and beyond. Only the first half of the table corresponding to a particular l is included because of symmetry, e.g. d^7 forms the same terms as d^3.

The data in /CONSTS/ defines commonly-used constants, whose precision could be changed if necessary.

CG(J1,J2,J3,M1,M2,M3)
is a real function that calculates Clebsch-Gordan coefficients. Factorials are required to be pre-defined in common block /FACT/ by a call to routine SHRIEK.

CHOP
is a routine taken from the angular momentum package [50,51]. It determines the shells whose interaction with all shells in the configuration arises purely as an average energy.

INTACT
is taken from the angular momentum package [50,51]. It sets the interaction energy between two shells whose orbital angular momentum is l and l', where l and $l' \leq 4$. The first term of this interaction energy is always $F^0(l,l')$, and this is not given in this routine.[5] Thus only the extra terms are produced here.

For equivalent electrons (argument `IEQUIV` = 1), there will be F^k integrals only. For non-equivalent electrons (`IEQUIV` = 2), there will be G^k integrals only.

The expressions for the interaction energies involving shells with $l \leq 3$ are given in Eqs. (14.20) and (14.22) of Slater [81]; in the last of his Eqs. (14.22) a term $-\frac{1}{14}G^0(f,f')$ is omitted - this is included in this routine. The interaction energies for g-electron shells may be evaluated using his Eqs. (13.12), (13.17), (14.19) and (14.21).

MEKEST
saves or restores the common block /MEDEFN/ and arguments `IRHO`, `ISIG`, `IRHOP`, `ISIGP`.

ORTHOG
is taken from the angular momentum package [50,51]. It checks for simple orthogonality of the configurations

[5] The direct and exchange radial integrals, $F^k(ab)$ and $G^k(ab)$ respectively, are defined in terms of the Slater radial integral by $F^k(ab) = R^k(abab)$ and $G^k(ab) = R^k(abba)$ for orbitals a and b.

```
5.6            CFP   CFPD
5.7                  CFPP
```

Fig. 19. Calling tree for CFP. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

due to coupling differences or uneven parity which would lead to zero matrix elements, The configurations are input in /MEDEFN/, and the routine argument LET is returned 0 if the two configurations will have zero Hamiltonian matrix element, or 1 if there is no obvious angular momentum orthogonality.

REDUCE
removes spectator ^{1}S shells which have no effect in angular or spin integrals. If a change in the common block /MEDEFN/ is to be made, its present situation is preserved by a call of routine MEKEEP. REDUCE calls routine MEKEST.

RME(l,l',k)
is a real function taken from the angular momentum package [50,51]. It evaluates the reduced matrix element using the formulae of Fano and Racah [45] chapter 14, p.81:

$$\mathrm{RME} = (l \,||\, C_k \,||\, l') = 0, \qquad \text{if } l+k+l' \text{ is odd}$$

otherwise

$$\mathrm{RME} = [(2l+1)(2l'+1)]^{1/2} \times \left[\frac{(l+l'-k)!(l'+k-l)!(k+l-l')!}{(2g+1)!}\right]^{1/2} \left[\frac{g!}{(g-l)!(g-l')!(g-k)!}\right]$$

where $2g = l+k+l'$. The orbital angular momenta l, l' of the interacting shells and order k are input as integer arguments. Factorials are required to be pre-defined in common block /FACT/ by a call to routine SHRIEK.

SETUPE
is a routine taken from the tensor operator package [67]. It generates the arrays defining the quantum numbers, occupation and coupling of the shells, for each of the two configurations involved in a single matrix element. The arrays are then in a form suitable for use in routine TENSOR. Only those shells occurring in at least one configuration are included.

7.2. CFP — Fractional Parentage Coefficients

This routine is called from the HOWTS, FANO and TENSOR packages, and is the controlling routine of the fractional parentage coefficient package (Allison [1], Chivers [32]). See Fig. 19 for a flow diagram.
BLOCK DATA — BDLIB2 defines the d-shell coefficients in /FRPAR2/;
CFPD is called for a d-shell coefficient [32];
CFPP is called if a p-shell coefficient is required [1];
For an f-shell (or higher) the program allows a maximum of two electrons in the shell (the result is in this case trivially unity) since f-shell fractional parentage coefficients have not been made available here.

7.3. Angular integrals (FANO,HOWTS,TENSOR)

FANO
This package, described by Hibbert [50,51], calculates angular and spin weighting factors for the two-electron

```
5.1   FANO   GENSUM DRACAH
5.2          J23ANG
5.3          J23SPN
5.4          KNJ
5.5          MODJ23
5.6          MUMDAD CFP   ----
5.8          NJGRAF ----
5.35         NTAB1
5.36         SETJ1
```

Fig. 20. Calling tree for FANO. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

contribution to the Hamiltonian matrix for a given initial and final configuration using the method of Fano [44]. See Fig. 20 for a flow diagram.

The Hamiltonian matrix element associated with the electron-electron interaction V can be written as in Eq. (4) of Hibbert [50,51] as

$$\langle \varphi_j \mid V \mid \varphi_{j'} \rangle = \sum_{\rho\sigma\rho'\sigma'} \Bigg[\sum_k a_k^{\rm d} R^k(n_\rho l_\rho, n_\sigma l_\sigma, n_{\rho'} l_{\rho'}, n_{\sigma'} l_{\sigma'}) + \sum_k a_k^{\rm e} R^k(n_\rho l_\rho, n_\sigma l_\sigma, n_{\sigma'} l_{\sigma'}, n_{\rho'} l_{\rho'}) \Bigg]$$

where R^k is the appropriate two-electron radial integral, and the ρ and σ indices label the interacting shells and are input as routine arguments. The ranges of k (which is incremented in steps of 2) for the direct ($\mathtt{KD1} \le k \le \mathtt{KD2}$) and exchange ($\mathtt{KE1} \le k \le \mathtt{KE2}$) integrals are determined in FANO and returned in common block /XATION/, which also returns the calculated angular and spin weighting factors in arrays AMULT and BMULT. These weighting factors are, for the direct and exchange integrals respectively:

$$\begin{aligned} a_k^{\rm d} = {} & N(l_\rho \,||\, C_k \,||\, l_{\rho'})(l_\sigma \,||\, C_k \,||\, l_{\sigma'}) \\ & \times [1 + (1-\delta_{\rho\sigma})(1-\delta_{\rho'\sigma'})][(2l_\sigma+1)(2l_{\rho'}+1)]^{-1/2} \\ & \times \sum\sum\sum\sum (cfp)(cfp)(cfp)(cfp)\, S(1)\, L_k(1) \end{aligned}$$

$$\begin{aligned} a_k^{\rm e} = {} & N(l_\rho \,||\, C_k \,||\, l_{\sigma'})(l_\sigma \,||\, C_k \,||\, l_{\rho'}) \\ & \times (2 - \delta_{\rho\sigma} - \delta_{\rho'\sigma'})[(2l_\sigma+1)(2l_{\sigma'}+1)]^{-1/2} \\ & \times \sum\sum\sum\sum (cfp)(cfp)(cfp)(cfp)\, S(2)\, L_k(2) \end{aligned}$$

where

$$N = \tfrac{1}{2}\,(-1)^{\Delta P}\,[N_\rho(N_\sigma - \delta_{\rho\sigma})N_{\rho'}(N_\sigma - \delta_{\rho'\sigma'})]^{1/2}$$

The various coefficients are evaluated by routine calls as follows:

J23SPN for $S(\epsilon, \epsilon')$, which denotes the spin recoupling coefficient in Eq. (35) of Fano [44];

J23ANG for $L_k(1)$ and $L_k(2)$, the two angular recoupling coefficients in Eq. (41) of Fano [44];

MUMDAD for the fractional parentage coefficients (cfp) from Fano's Eqs. (24) or (26), calculated by calls to the CFP package;

RME for the reduced matrix elements $(l \,||\, C_k \,||\, l')$.

```
6.1  HOWTS  CFP    ----
6.2         NJGRAF ----
6.3         SETM
```

Fig. 21. Calling tree for HOWTS. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

```
13.1  TENSOR CFP    ----
13.2         NJGRAF ----
13.3         NTAB1
13.4         TRITST
```

Fig. 22. Calling tree for TENSOR. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

HOWTS

This package, described by Hibbert [52], calculates the angular and spin weighting factors for the one-electron contribution to the Hamiltonian matrix for a given initial and final configuration, when these are different.

See Fig. 21 for a flow diagram.

$$\langle \varphi_j \mid H_0 \mid \varphi_{j'} \rangle = \sum_{\sigma\sigma'} x(\sigma,\sigma')\, Q(n_\sigma l_\sigma, n_{\sigma'} l_{\sigma'})\, \delta_{l_\sigma l_{\sigma'}}$$

where x is the angular and spin integral calculated in routine HOWTS and Q the appropriate one-electron radial integral; the σ indices label the interacting shells and are input as routine arguments. The weighting factor is returned as an argument (TIMES).

SETM sets constants in /MVALUE/ used in inner routines.

TENSOR

is a package for the evaluation of angular and spin factors in the reduced matrix element of any one-electron tensor operator between arbitrarily coupled LS configurations. This package has been published by Robb [67].

See Fig. 22 for a flow diagram.

The reduced matrix element is expressed in the form

$$(\alpha LS \,\|\, \sum_n T_k(n) \,\|\, \alpha' L' S') = \begin{cases} \text{VSHELL(1)} * (l_\rho \,\|\, T_k \,\|\, l_\sigma) & \text{for } \rho \neq \sigma \\ \sum\limits_{\text{IRHO}=1}^{\text{IHSH}} \text{VSHELL(IRHO)} * (l_\rho \,\|\, T_k \,\|\, l_\sigma) & \text{for } \rho = \sigma \end{cases}$$

where $T_k(n)$ is a tensor operator of order k which operates on the nth electron, and is *either* independent of spin *or* completely spin-dependent (in which case $k = 1$ and $l_\rho = l_\sigma = \frac{1}{2}$). ρ and σ denote the interacting shells (IRHO and ISIG). Routine TENSOR calculates VSHELL and the remaining angular contribution is evaluated in function RME. Routine SETUPE must be called prior to entering TENSOR to set up the arrays in common block /MEDEFN/.

These reduced matrix elements are defined through the Wigner-Eckart theorem as:

$$\langle \alpha LSM_L M_S \mid \sum_n T_k^m(n) \mid \alpha' L' S' M_L' M_S' \rangle = \delta_{SS'} \delta_{M_S M_S'} \frac{C(L'kL; M_L' m)}{(2L+1)^{1/2}} \times (\alpha LS \,\|\, \sum_n T_k(n) \,\|\, \alpha' L' S')$$

for the spin-independent case; a similar relation holds for the spin-dependent case. The $C(L'kL; M_L' m)$ are Clebsch-Gordan coefficients.

```
7.1  HSLDR  EIGEN
7.2         EIGVEC NORM
7.3         HOUSE
7.4         VECTOR BAKSUB
7.5                NORM
```

Fig. 23. Calling tree for HSLDR. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

TRITST(l_1, l_2, l_3) is a real function taken from the tensor operator package [67]. It tests whether or not three angular momenta values satisfy a triangular relationship.

NTAB1(l, q)
This integer function is called by the FANO and TENSOR packages and returns the row of NTAB (defined in /TERMS/ in BLOCK DATA) where may be found the parents of a shell given by l^q. For example, the row of NTAB containing the parents of one of the p^4 terms ($l = 1$, $q = 4$) is that containing the p^3 terms; thus NTAB1=4. Use is made of the fact that the list of possible parents is symmetrical about the configuration $l^{(2l+1)}$. For one electron in a term, the parent is always a ^{1}S term; otherwise the value of NTAB1 depends on the l value of the electrons. The function is aborted if there are more than two $l \geq 3$ electrons in a shell.

7.4. HSLDR — matrix diagonalization

This package diagonalizes a real symmetric matrix using the Householder method [84] to find all the eigenvalues and eigenvectors. See Fig. 23 for a flow diagram. The routine works entirely in memory and is economic, requiring little more space than that required to store the upper triangle of the matrix, which is input in a linear array of dimension LENGTH = $(n * (n + 1))/2$, where n is the order.

Diagonalization proceeds in two stages: firstly a tri-diagonalization reduction is performed; secondly the eigenvalues and eigenvectors are formed from the tri-diagonal form. All the eigenvalues and the first eigenvector are found in the first call to HSLDR; the remaining eigenvectors are calculated one at a time in each subsequent call to HSLDR.

Tri-diagonalization is performed by a Householder reduction of a matrix A of order n, effected by $n - 2$ similarity transformations with orthogonal matrices $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, ... $\boldsymbol{P}_{n-2}$ successively [84]:

$$\boldsymbol{A}^{(r+1)} = \boldsymbol{P}_r \boldsymbol{A}^{(r)} \boldsymbol{P}_r, \qquad r = 1, 2, \ldots n-2 \tag{122}$$

where $\boldsymbol{A}^{(1)}$ is the original $n \times n$ matrix $\boldsymbol{A}$. At each step a row and a column is set to zero (except for the diagonal and adjacent element). Clearly this is recursive, but within one cycle of the reduction the transformation acts on rows independently.

Consider a cycle at step r. Matrix $\boldsymbol{A}^{(r)}$ consists of $r - 1$ rows and columns already tri-diagonalized, with a square submatrix starting in the rth row and column remaining to be transformed. First sum the squares of the off-diagonal elements down the rth column (or row - the matrix is symmetric) of $\boldsymbol{A}^{(r)}$ and form:

$$S^2 = \sum_{i=r+1}^{n} A_{ir}^2, \qquad \mathrm{BKR} = S(S \pm A_{r+1,r})$$

(the sign is arbitrary and is chosen to avoid subtraction errors). Then define a column vector $\boldsymbol{u}$ with zeros in the first $r + 1$ elements and the remaining elements taken from the rth column of $\boldsymbol{A}^{(r)}$:

$$\boldsymbol{u}^{\mathrm{T}} = (0, \ldots 0, A_{r+1,r} \pm S, A_{r+2,r}, \ldots A_{nr})$$

which implies that $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}$ = 2BKR. It can easily be shown that $\boldsymbol{P}_r = \boldsymbol{I} - \boldsymbol{u}.\boldsymbol{u}/\mathrm{BKR}$ is an orthogonal matrix (i.e. $\boldsymbol{P}_r^{\mathrm{T}}\boldsymbol{P}_r = \boldsymbol{I}$). Using this form of $\boldsymbol{P}_r$ in Eq. (122) and defining:

$$p = A^{(r)}u/\mathrm{BKR}, \qquad K = u^{\mathrm{T}}p/\mathrm{BKR}, \qquad q = p - \tfrac{1}{2}Ku \tag{123}$$

then

$$A^{(r+1)} = A^{(r)} - uq^{\mathrm{T}} - qu^{\mathrm{T}} \tag{124}$$

This transformation of $A^{(r)}$ ensures that the first $r-1$ rows and columns (which have already been tri-diagonalized in previous cycles) remain unaltered, the rth diagonal element remains the same and its adjacent element becomes $\pm S$, and the rest of the rth row and column is made zero.

In Eq. (123), since u and q are both column vectors, the evaluation of q and the scalar quantity K are single-loop vector operations. However, writing out the evaluation of p in element form:

$$p_i = \left(\sum_{j=r+1}^{i} A_{ji}u_j + \sum_{j=i+1}^{n} A_{ij}u_j \right) /\mathrm{BKR} \tag{125}$$

two sums are required because the matrix A is stored in upper triangular form (i.e. the lower triangle is not defined. Both inner (j) and outer (i) loops show parallelism, as does the evaluation of Eq. (124):

$$A_{ij} = A_{ij} - u_i q_j - q_i u_j, \qquad i,j = r+1, n \tag{126}$$

Note that the tri-diagonalization process consists of three levels of nested loops (i.e. it is an 'n^3 process' - computing time normally increasing with the cube of the order of matrix n); only the outer loop (over r in Eq. (122)) is recursive, the inner loops are usually vectorisable on suitable computers.

Let us now summarise the routines involved in order to calculate eigenvalues and eigenvectors.

BAKSUB is called by VECTOR to solve for X in the eigenvector equation $UX = B$ by back substitution.

EIGEN finds the eigenvalues of the tri-diagonal matrix (these are the same as the eigenvalues of the original matrix) using the Sturm sequence property, applying a bisection method to determine eigenvalues to a specified accuracy.

The polynomials $p_1(x)$, $p_2(x)$, $\ldots p_n(x)$ are a Sturm sequence in the interval (a,b) if

- $p_n(x)$ has a constant sign in (a,b);
- at any zero of $p_k(x)$ in (a,b), $p_{k-1}(x)$ and $p_{k+1}(x)$ are non-zero and of opposite sign ($k = 2, \ldots n-1$).

The sequence is generated by setting up the matrix $\lambda I - B$ where none of the co-diagonal elements in B are non-zero. Let $f_i(\lambda)$ denote the determinant of the leading principal minor of $\lambda I - B$. Using determinantal relations it can be shown that

$$f_i(\lambda) = (-\alpha_i + \lambda)f_{i-1}(\lambda) - \beta_{i-1}^2 f_{i-2}(\lambda).$$

By starting with $f_0(\lambda)$=1 and $f_1(\lambda)$=$\lambda - \alpha_1$, then the full sequence can be generated using the above relations. It can be shown that the zeros of $f_i(\lambda)$ separate those of $f_{i-1}(\lambda)$ and hence that a Sturm sequence has been generated. An important property of a Sturm sequence is the if $V(\lambda)$ is the number of changes of sign in the sequence, then $V(\lambda)$ gives the number of eigenvalues in excess of λ. With this information, bisection and the subsequent location of the eigenvalues can quickly proceed as follows: if we require λ_k in the interval (a,b), then choose $c = (a+b)/2$ and if $V(c) > k$, then take $a = c$; otherwise, take $b = c$. This procedure is continued until a and b differ by less than a pre-assigned tolerance (EPSI, defined in the call to HSLDR), when a value for λ_k can finally be allocated.

EIGVEC then obtains the eigenvectors of the original matrix using details of the matrices used in transforming the original matrix A to tri-diagonal form. Thus if x is an eigenvector of the tri-diagonal matrix the corresponding eigenvector v of matrix A is given by:

$$v = P_1 P_2 \ldots P_{n-2} x$$

```
5.8       NJGRAF BUBBLE DELTA
5.9              CUT1L  DELTA
5.10                    PRINTJ
5.11                    ZERO   PRINTJ
5.12             CUT2L  DELTA
5.13                    PRINTJ
5.14             CUTNL  DELTA
5.15                    PRINTJ
5.16             DIAGRM INTAB
5.17                    PRINTJ
5.18                    WAY
5.19             GENSUM DRACAH
5.20             LOLPOP DELTA
5.21             ORDTRI CHANGE
5.22                    PRINTJ
5.23             POLYGN PRINTJ
5.24             PRINTJ
5.25             SEARCH CHANGE
5.26                    PRINTJ
5.27             SETDM
5.28             SETTAB DELTA
5.29                    PRINTJ
5.30             SPRATE CHVAR
5.31                    VAR
5.32             SQUARE
5.33             TRIANG TRDEL
5.34             ZERO   PRINTJ
```

Fig. 24. Calling tree for NJGRAF. The routines are given in alphabetical order within each branch **not** the order in which they are actually called.

The vector $\boldsymbol{v}$ is calculated from the relations

$$\boldsymbol{x}_r = \boldsymbol{P}_r \boldsymbol{x}_{r+1}, \qquad r = n-2, \dots, 1$$

where $\boldsymbol{x}_{n-1} = \boldsymbol{x}$ and $\boldsymbol{x}_1 = \boldsymbol{v}$, so that the product of the $\boldsymbol{P}_r$ does not have to be formed explicitly. Also advantage is taken of accumulating a scalar product when calculating $\boldsymbol{P}_r \boldsymbol{x}_{r+1}$:

$$\boldsymbol{P}_r \boldsymbol{x}_{r+1} = (\boldsymbol{I} - 2\boldsymbol{u}_r \boldsymbol{u}_r^{\mathrm{T}}) \boldsymbol{x}_{r+1} = \boldsymbol{x}_{r+1} - 2(\boldsymbol{u}_r^{\mathrm{T}} \boldsymbol{x}_{r+1}) \boldsymbol{u}_r$$

HOUSE performs the tri-diagonalization by Householder's method as described above.

NORM is called from VECTOR and EIGVEC to renormalise the vector $\boldsymbol{x}$ and $\boldsymbol{v}$ such that the largest component is unity.

VECTOR finds the eigenvectors of the tri-diagonal matrix by inverse iteration. Calls BAKSUB.

7.5. NJGRAF — recoupling package

This general recoupling package is described by Bar-Shalom and Klapisch [5], and is called from the FANO, HOWTS and TENSOR packages. See Fig. 24 for a flow diagram.

The recoupling coefficient takes the form (Burke [20]) $(j_1, j_2 \dots, j_n, j \mid j_1, j_2 \dots, j_n, j)$, where the n values $j_1 \dots j_n$ can be coupled in an arbitrary and in general different way in the bra (initial state) and ket (final state) vectors to form the total angular momentum j. The efficient method of Bar-Shalom and Klapisch uses graphical analysis to construct the formula which involves summing products of Racah coefficients.

GENSUM
is called by the FANO and NJGRAF packages. It evaluates the product of the $(2j+1)^{1/2}$ factors, the

$(-1)^j$ factors, the $(-1)^{-j}$ factors and Racah coefficients, which are input in arrays K6, K7, K8 and KW as routine arguments, and carries out the required summation subject to the restrictions imposed by the triangular conditions. GENSUM calls the routine DRACAH.

DRACAH
evaluates a Racah coefficient $W(abcd;ef)$ (Fano and Racah [45]). The routine employs the algorithm described by Scott and Hibbert [72]. It works for integer and half-integer values of angular momenta, which are input in the routine argument list as integers representing twice the actual values. Each of the four triads $(abe),(cde),(acf),(bdf)$ has an integral sum. Moreover, the Racah coefficient satisfies selection rules such that each of these four triads must form a possible triangle, i.e. must satisfy the condition that any side of a triangle is smaller than or equal to the sum of the other two sides; DRACAH returns $W = 0$ if any triad does not form a triangle.

Logs of factorials are required to be pre-defined in common block /FACTS/ by a call to routine FACTT.

FACTT
calculates logs of factorials, $\texttt{FACT}(n+1) = \ln(n!)$ which are stored in common block /FACTS/ and used in the Racah coefficient routine DRACAH.

8. Preprocessing

8.1. Introduction

Dimensions in RMATRX1 are set by preprocessing parameters. These parameters begin with an ampersand (&), and must be substituted with numbers in order to produce a FORTRAN compilable source. They are mainly confined to PARAMETER statements of the form

```
      PARAMETER (MZCHF=&CHF)
```

at the start of the relevant routines. A convention is that the assigned variable names begin with MZ. A number of other variables (with names beginning with MX) are derived from these in subsequent PARAMETER statements. All common blocks can be found in the PROGRAM routine of each module and these dimension setting PARAMETER statements are located there also. The &-parameters have been kept to a near minimal set for convenience.

There are two suggested ways for carrying out the preprocessing
- using the FORTRAN program PREP
- using a text editor such as the UNIX stream editor sed

8.2. PREP program

A preprocessor program is provided: PREP, written in standard FORTRAN (Day [40]). PREP is not however essential to the R-matrix package - it merely performs the parameter substitution for dimensions, and is useful in batch runs.

The preprocessor is common to a number of other programs, such as the Opacity Project version of the R-matrix programs (Berrington et al. [11]) and the no-exchange programs (Burke et al. [29]), and similar preprocessing parameters are used. PREP requires four files:
unit 1 the original version of one of the R-matrix modules (input);
unit 2 the FORTRAN version of the R-matrix module (output);
unit 3 a scratch file;

unit 4 user supplied input consisting of two columns on each line using FORMAT(A4,I4), the first column being the preprocessing parameter (which begins with '&'), the second column being the number to substitute, as described below.

8.3. &-parameters

The following is a description of the 16 preprocessing parameters used (with some suggested values in square brackets).

&CHF [25] the highest number of coupled channels (max(`NCHAN`)), i.e. the range of index i in the summation of Eq. (12) and Eq. (27).

&CHL [25] the highest number of coupled channels in LS-coupling.

&FAC [32] largest factorial available. Factorials are calculated in routines SHRIEK and FACTT.

&IPH [2] = 1 for electron scattering only, = 2 otherwise (`IPOLPH`).

&LMX [4] maximum number of multipoles in the long-range potential (`LAMAX`), i.e. the range of λ in Eq. (30).

&LR1 [5] highest $l+1$ for bound orbitals (`LRANG1`).

&LR2 [20] highest $l+1$ for continuum orbitals (`LRANG2`).

&MEG [1] megawords of memory required in each module – in the case of optional use, insufficient memory will result in the opening and use of scratch files on disk, with consequent increase in I/O overheads. However, the results should be correct.

- **STG1** needs sufficient space to store two-electron radial integrals associated with a particular l and l' combination (the number of such integrals can be obtained from a dimension test run of STG1 by setting `NBUG7=1` in the input data);

- **STG2** needs sufficient space to store two-electron radial integrals associated with a particular l and l' combination – optional use is to store **all** radial integrals in memory, and also to store Hamiltonian matrix blocks for sorting in routine SETMX1;

- **RECUPD** has no compulsory need – optional use is to store continuum-continuum matrix element blocks in the $J\pi$ representation while they are being recoupled, and also to store recoupling coefficients;

- **STGH** has no compulsory need, and no requirement at all for electron scattering calculations (`IPOLPH=1`) – optional use is to store the Hamiltonian matrix eigenvectors, and the dipole matrices while they are being transformed;

- **STG4** needs sufficient space for storing the entire H file in memory, and in the case of photoionization for storing vectors containing bound state data; in order to estimate how much memory is required in STG4, it should be noted that the largest items are normally the surface amplitudes Eq. (20) on the H file: these are of dimension `NCHAN*MNP2`, where `MNP2=NCHAN*NRANG2+NCFGP`, and there is one such array for each $(N+1)$-electron symmetry;

In order to run the codes on a PC with a small memory the variable `MZKIL` was introduced. This sets the number of kilowords of memory and is used along with &MEG so that the total memory allocation is (&MEG*1000+`MZKIL`) kilowords. This variable has been set to 100 in all modules using the statement

```
PARAMETER (MZKIL=100)
```

&NC1 [50] highest number of N-electron target configurations for given symmetry.

&NC2 [100] highest number of $(N+1)$-electron configurations for given symmetry (max(`NCFGP`)), i.e. the range of index j in the second summation of Eq. (12).

Table 1
The modules in which the preprocessing variables are used

Variable	STG1	STG2	RECUPD	STGH	STG4	STGLIB	CREES
CHF		●	●	●	●		●
CHL			●				
FAC	●	●			●	●	
IPH			●				
LMX	●	●	●	●	●		●
LR1	●	●	●			●	
LR2	●	●	●	●		●	
MEG	●	●	●	●	●		
NC1		●	●				
NC2		●	●	●		●	
NPT	●						
NR1	●	●	●			●	
NR2	●	●	●	●			
OCC		●	●			●	
SLP		●	●	●	●		
TAR		●	●	●	●		

&NPT [800] the number of radial mesh points in the inner region.
&NR1 [5] highest n for bound orbitals.
&NR2 [20] number of continuum orbitals for a given l (NRANG2).
&OCC [15] maximum number of occupied shells in a given configuration.
&SLP [80] number of different $(N+1)$-electron symmetries.
&TAR [100] number of N-electron configurations or target states (max(NCFG, NAST)).

The modules in which the preprocessing variables are used is summarised in Table 1 with ● indicating its use.

8.4. Using a text editor

Rather than use PREP it may be more convenient to use a text editor.

As an example of using a text editor to preprocess a code, consider the UNIX stream editor, sed. If the unprocessed code is `stg1.pre` and the sed commands are in file `stg1.sed` then you would issue the following command :

```
sed  -f stg1.sed  stg1.pre  > stg1.f
```

The processed output is piped to file `stg1.f`. This can be compiled in the normal way.

The format of the sed commands is for example

```
s/&NPT/800/g
```

which is a global (i.e. , the ending g) substitution of the text &NPT by 800. This would correspond to setting the number of radial mesh points in module STG1 to 800.

An example of a sed (`.sed`) file is given in Fig. 25.

8.5. Memory requirements

The dimensions given in Fig. 25 were used to compile the modules on a Cray Y-MP EL. The resulting memory requirements are given in Fig. 26.

Module RECUPD has large memory requirements when the number of channels and continuum orbitals is increased.

```
#
# Sample .sed file for RMATRX1. (comments begin with #)
#
# THE FOLLOWING DIMENSION PLANTS HAVE BEEN MADE :
#
#   CHF (25)   NUMBER OF SCATTERING CHANNELS                   (NCHAN)
#   CHL (25)   NUMBER OF CHANNELS IN LS COUPLING               (NCHAN)
#   FAC (32)   LARGEST FACTORIAL AVAILABLE ON MACHINE
#   IPH (2)    =1 FOR ELECTRON SCATTERING ONLY, =2 OTHERWISE (IPOLPH)
#   LMX (4)    MULTIPOLES IN POTENTIAL                         (LAMAX)
#   LR1 (5)    HIGHEST L+1 FOR BOUND ORBITALS                  (LRANG1)
#   LR2 (20)   HIGHEST L+1 FOR CONTINUUM ORBITALS              (LRANG2)
#   MEG (1)    MEGA-WORDS OF MEMORY,
#                FOR INTEGRAL STORAGE,
#                TO REDUCE DISK I/O,
#                TO STORE H FILE,
#                TO STORE BOUND STATE WAVEFUNCTION AND DIPOLE VECTORS
#   NC1 (50)   TARGET N-ELECTRON CONFIGURATIONS FOR GIVEN SYMMETRY
#   NC2 (100)  N+1 ELECTRON CONFIGS FOR GIVEN SYMMETRY         (NCFGP)
#   NPT (800)  RADIAL TABULAR POINTS                           (IRX(NIX))
#   NR1 (5)    HIGHEST N FOR BOUND ORBITALS                    (MAXNHF(L))
#   NR2 (20)   NUMBER OF CONTINUUM ORBITALS FOR GIVEN L        (NRANG2)
#   OCC (15)   OCCUPIED SHELLS IN A GIVEN CONFIGURATION
#   SLP (80)   NUMBER OF DIFFERENT N+1 ELECTRON SYMMETRIES     (INAST)
#   TAR (100)  TARGET STATES OR CONFIGURATIONS                 (NAST,NCFG)
#
#   KIL (100)  KILO-WORDS OF MEMORY
#   You can use the KIL parameter to set kilo-words of memory.
#   At present this is set to 100. To reset the variable to say 500
#   you can use:
#        s/MZKIL=100/MZKIL=500/g
#
s/&CHF/25/g
s/&CHL/25/g
s/&FAC/32/g
s/&IPH/2/g
s/&LMX/4/g
s/&LR1/5/g
s/&LR2/20/g
s/&MEG/1/g
s/&NC1/50/g
s/&NC2/100/g
s/&NPT/800/g
s/&NR1/5/g
s/&NR2/20/g
s/&OCC/15/g
s/&SLP/80/g
s/&TAR/100/g
```

Fig. 25. Sample sed file for the codes.

```
STG1        1.9346 MWords
STG2        1.6880 MWords
RECUPD      2.3594 MWords
STGH        1.5488 MWords
STG4        1.4248 MWords
```

Fig. 26. Module memory requirements. These use the dimensions of Fig. 25.

9. GLOSSARY: detailed description of input/output data

Appended in brackets ( ) to the name of an array is an indication of its dimensionality.

Appended in braces { } are any recommended or default values.

Appended in square brackets [ ] is a module name if the scope is restricted to one particular program module.

ABUTL,ABUTV(k,i) :

Buttle correction dipole matrix elements - in length and velocity forms - between basis functions and channels of each $(N+1)$-electron state involved in the dipole transition; see Eq. (91).

AC(i,i') $= (\overline{\Phi}_i \,||\, \hat{r} \,||\, \overline{\Phi}_{i'})$; $i = 1,$NCHAN; $i' = 1,$MCHAN :

required for the outer region contribution to the dipole matrix. See Section 1.3.2 and Eq. (92).

AC {0.005} [STG4] : accuracy for the Crees asymptotic package, ASYPCK. Variable EROR is used within ASYPCK.

AIJ(NAST,j); $j = 1,$NTCON :

the weighting of the jth configuration (in the order generated by the CONFIG package or read in from JREAD) for each target state.

ALPHL,ALPHV [STG4] :

frequency dependent dipole polarizability - in length and velocity forms - calculated using Eq. (121).

B(NAST,k); $k = 1,$JNTCON [RECUPD] :

the weighting factor of the kth configuration of each target level. These are equivalent to the coupling coefficients of Eq. (105) if JRELOP(3) = 1, or to the term-coupling coefficients of Eq. (109) if JRELOP(3) = 0.

BBUTL,BBUTV(k,i) :

Buttle correction dipole matrix elements - in length and velocity forms - after multiplying by the eigenvectors in STGH, between basis functions and channels of each $(N+1)$-electron state involved in the dipole transition; see Eq. (91).

BETALR,BETAVR [STG4] : β asymmetry parameter (real) - in length and velocity forms - calculated using Eq. (117).

BETALI,BETAVI [STG4] : (Im$\beta = 0$, for checking purposes).

BLC(i,i') $= (\overline{\Phi}_i \,||\, \mathbf{R} \,||\, \overline{\Phi}_{i'})$; $i = 1,$NCHAN; $i' = 1,$MCHAN :

required for the outer region contribution to the dipole length matrix. See Section 1.3.2 and Eq. (92).

BNORM(LRANG2) : dipole normalisation involving Buttle term.

BSTO {0.0} : the value of the logarithmic derivative to be imposed on the continuum orbitals $u_{ij}(r)$, i.e. Eq. (16).

BVC(i,i') $= (\overline{\Phi}_i \,||\, \frac{\mathrm{d}}{\mathrm{d}r} \,||\, \overline{\Phi}_{i'})$; $i = 1,$NCHAN; $i' = 1,$MCHAN :

required for the outer region contribution to the dipole velocity matrix. See Section 1.3.2 and Eq. (93).

C(NCO) [STG1] :

Slater-type orbital coefficients in Eq. (70). The coefficients C(J) of the expansion can be supplied in Clementi form [34], which differ from Eq. (70) by a factor

$$\mathcal{N} = \frac{\sqrt{2 * \mathrm{IRAD(J)}!}}{(2 * \mathrm{ZE(J)})^{(\mathrm{IRAD(J)}+\frac{1}{2})}}$$

However, the program switches automatically between the two by checking the normalisation of the orbital. It is the Slater-type form which is actually used in STG1.

CF(i,j,λ) $= 2a_{ij}^{\lambda}$; $i = 1,$NCHAN; $j = 1,$NCHAN; $\lambda = 1,$LAMAX :

asymptotic coefficients of multipole order $\lambda + 1$, as in Eq. (29) and Eq. (88).

CGC(m) $= (2L+1)^{-1/2}C(L'1L;m0)$; $m = 1,$MAXM1 :

Clebsch-Gordan coefficient divided by the square-root factor occurring in the definition of the reduced matrix element for each angular momentum L. See Section 1.3.2, and Eq. (94).

COEFF(3, $l+1$); $l+1$ = 1, LRANG2 :
holds the three parameters for the Buttle correction fit for each continuum angular momentum l. Seaton's fitting procedure [77] is normally used as described in routine NEWBUT in STG1. See Eq. (67).

CPOT(NPOT) [STG1] : the zero-order potential energy function in Eq. (79).

DEL, DEV(λ, λ') = $(\phi_\lambda \| D^{(1)} \| \phi_{\lambda'})$:
the reduced dipole matrix elements - in length and velocity forms - between the R-matrix basis, where the ϕ are collectively the basis functions. See Eq. (48) and the description of routine DMEL is STG2. This matrix is normally input and output in blocks, so the array handling the I/O does not have to be fully dimensioned.

DELTA {0.00002} [STG1] : the energy increment for use in the BASFUN package.

DEOPEN [STG4] : interval in energy when all channels are open.

DL, DV(NCHAN) = $\langle \Psi_j^- \| M \| \Psi_0 \rangle$ [STG4] :
dipole vector elements as defined in Eq. (120) - in length and velocity forms.

DML, DMV(k, k') = $(\psi_k \| D^{(1)} \| \psi_{k'}) = V_k^{\mathrm{T}} D V_{k'}$:
the reduced dipole matrix elements - in length and velocity forms - between the ψ_k basis. See Eq. (49) and the description of routine DMAT in STGH.

DQN [STG4] : interval for effective principal quantum number n.

DUJ(NPTS, NBOUND) : values of one-electron operator terms $Q_{nl}(r)$ of Eq. (72).

E [STG4] : electron energy in Ry.

E0 [STG4] : first energy (in Ry if IRAD = 0, or z^2-scaled Ry if IMESH = 1; and refers to the energy of the incident electron if IRAD = 0 or 1, or photon if IRAD = 2 or 3).

E1 [STG4] : initial state energy in Ry relative to target ground state.

EF [STG4] : final bound state energy for gf-value.

EFFN [STG4] : effective principal quantum number of the $(N+1)$-electron bound state relative to N-electron target ground state.

EIG(i) [STG4] : eigenphase in ith channel.

EIGENS(NRANG2, $l+1$); $l+1$ = 1, LRANG2 :
k_{ij}^2 eigenvalues in Ry for continuum functions of angular momentum l_i; see Eq. (15).

EINCR [STG4] : energy interval (in Ry if IRAD = 0, or z^2-scaled Ry if IMESH = 1; and refers to the energy of the incident electron if IRAD = 0 or 1, or photon if IRAD = 2 or 3).

EMESH [STG4] : energy mesh (in z^2-scaled Ry).

EMIN, EMAX [STG4] : minimum, maximum electron energy (in z^2-scaled Ry).

EN : eigenvalue of the N-electron Hamiltonian matrix.

ENAT(NAST) : the total energy of each target state in a.u. .
If supplying level energies as user input to module RECUPD, then if:
JRELOP(3) = 0 the energies should give the observed splittings;
JRELOP(3) = 1 the energies should be the theoretical ones.

ENDS(NRANG2 + 1, LRANG2) = u_{ij}(RA) :
boundary amplitudes. The NRANG2 + 1 location is reserved for Buttle amplitudes.

EP [STG4] : photon energy in Ry.

EST(NEST) [STGH] : observed target energies, in a.u. if EST(1) > 0, in Ry if EST(1) = 0, or expressed as cm^{-1} wave number if the values of EST are prefaced by a minus sign. This facility corrects the diagonal elements of the target Hamiltonian and yields correct channel energies in the asymptotic modules. The ordering of the target energies in this array must correspond exactly with the ordering of the target states defined by the user in module STG2 or RECUPD. See Section 5.4.

ETA {0.00002} [STG1] :

the accuracy parameter to which the continuum basis eigenvalues k_{ij}^2 are to be computed.

ETOT [STG4] : incident electron energy in Ry.

FL,FV [STG4] : gf-value - in length and velocity forms - calculated using Eq. (118).

H(LENGTH) : contains upper triangle of N-electron Hamiltonian matrix for a given target LS symmetry, stored in rows as a one-dimensional array from routine BOUND in module STG2. Stored on output file ITAPE3 for use in routine BOUNDJ in RECUPD.

HINT [STG1] : the basic integration step-length, see Eq. (77).

HNP1(MNP2,MNP2) : contains the $(N+1)$-electron Hamiltonian matrix blocks.

I1= 1 to indicate that initial state is the lowest state of that symmetry.

IBASSH(m,i); $i=1$,MAXORB; $m=1$,NOPTN [STG2] :

the number of electrons in the ith shell defining the mth basic configuration. The configurations retained will be the union of those generated by each basic configuration, subject to the overall restriction of MNAL and MXAL.

IBBI : the number of bound-bound multipole integrals stored in the RKSTO2 array.

IBBPOL($l+1,l'+1,\lambda$); $l+1=1$,LRANG1; $l'+1=1$,LRANG1; $\lambda=1$,LAMIND :

the position in the RKSTO2 array where the first bound-bound multipole integral corresponding to (l,l') is stored (with $l<l'$). Only those multipole values λ allowed by the triangle relations are stored.

IBC {0} [STG1] :

= 0 normally, in this case the boundary radius RA will be automatically calculated and a value BSTO = 0 chosen for the logarithmic derivative of the continuum functions at RA;

> 0 to read in the values of RA and BSTO (record 13);

> 1 to read the integration mesh parameters also (records 14).

IBCPOL($l+1,l'+1,\lambda$); $l+1=1$,LRANG1; $l'+1=1$,LRANG2; $\lambda=1$,LAMIND :

the position in the RKSTO2 array where the first bound-continuum multipole integral corresponding to (nl) bound, $(n'l')$ continuum, is stored (with $l<l'$). Only those multipole values λ allowed by the triangle relations are stored.

IBUG1...9 {0} : debug parameters. See module descriptions.

ICCPOL($l+1,l'+1,\lambda$); $l+1=1$,LRANG2; $l'+1=1$,LRANG2; $\lambda=1$,LAMIND :

the position in the RKSTO2 array where the first continuum-continuum multipole integral corresponding to (l,l') is stored (with $l<l'$). Only those multipole values λ allowed by the triangle relations are stored.

ICHECK {1} [RECUPD] :

= 0 to read in fine-structure energies and CI coefficients;

= 1 for automatic calculation of energies and CI coefficients by diagonalization of the target Hamiltonian in routine BOUNDJ.

ICODE : code version on output files.

ICOPY,ITOTAL {0,999} : a restart facility for STG2.

ICOPY is the position number of the last block of data to be copied from ITAPE2 to ITAPE3. It is not used in STG1.

ITOTAL is the total number of data blocks required on ITAPE3.

In STG2 each $(N+1)$-electron symmetry gives rise to four blocks of data on the output file: continuum-continuum, continuum-bound, bound-bound Hamiltonian matrix blocks, together with the associated asymptotic coefficients.

ICT((I1 + I2) * LRANG1 * LRANG1) :

for the direct continuum-continuum two-electron radial integrals,

the first I1 * LRANG1 * LRANG1 locations indicate the position in the RKSTO2 array where the first direct two-electron integral corresponding to the $(l_1l_2l_3l_4k)$ combination is stored (I1 = min(2*LRANG1 − 1, L+LP+1));

the second I2 * LRANG1 * LRANG1 locations indicate the position in the RKSTO2 array where the first exchange two-electron integral corresponding to the $(l_1l_2l_3l_4k)$ combination is stored (I2 = min(LRANG1+L, LRANG1+

LP)).

ICTBB(l_1+1, l_2+1, l_3*LRANG1$+l_4+1$); l_1+1; l_2+1; l_3+1; l_4+1 = 1,LRANG1 :
for the bound-bound two-electron radial integrals, the position in the ISTBB1 array where the first and smallest k value associated with the particular $(l_1l_2l_3l_4)$ combination is stored. Further allowed k values are stored consecutively. The corresponding value in the ISTBB2 array gives the position in the RKSTO1 array where the first two-electron integral corresponding to the $(l_1l_2l_3l_4k)$ combination is stored.

ICTBC(l_1+1, l_2+1, l_4*LRANG1$+l_3+1$); l_1+1; l_2+1; l_3+1 = 1,LRANG1; l_4+1 = 1,LRANG2 : for the bound-continuum two-electron radial integrals, the position in the ISTBC1 array where the first and smallest k value associated with the particular $(l_1l_2l_3l_4)$ combination is stored. Further allowed k values are stored consecutively. The corresponding value in the ISTBC2 array gives the position in the RKSTO2 array where the first two-electron integral corresponding to the $(l_1l_2l_3l_4k)$ combination is stored.

ICUT {0} [STG2] : is the total number of configurations generated or read if not all of them are to be stored. ICUT applies to the $(N+1)$-electron states and allows you to exclude configurations from their description. This facility is only for the experienced user. See array IKIP.

IDIAG {1} : NOT USED.

IDISC1,IDISC2,IDISC3,IDISC4 {8,9,10,11} : scratch files. See module descriptions.

IHX(I); I = 1,NIX [STG1] :
the multiple of the basic step length defining the integration step in the Ith interval, where IHX(I) = 2 * IHX($I-1$) with IHX(1) = 1.

IJNAST [RECUPD] : the total number of $(N+1)$-electron symmetries to be considered.

IKEY [STG2] : similar to NKEY, but for the $(N+1)$-electron configurations. (The IKEY = 2 option is not normally used, but if it is, the $(N+1)$-electron configurations are read from JREAD)

IKIP(I); I = 1,NCUT [STG2] :
= 0 if the Ith configuration is not to be stored;
= 1 if the Ith configuration is to be stored.
See the description of NCUT. This array is also used with ICUT.

IL [STG4] : L for selected symmetry (= $2J$ in intermediate-coupling).

IL1, IL2 [STG4] : L for initial, final bound state (= $2J$ in intermediate-coupling).

ILRGL : the initial L or $2J$ value for an $(N+1)$-electron transition on the D file.

IMESH [STG4] :
= 1 to generate an equal energy interval mesh, in z^2-scaled Ry;
= 2 to generate an equal effective n energy mesh;
= 3 to read energy mesh;
= 0, as 1 above, but energies in unscaled Ry:
electron energies if IRAD = 0 or 1, photon energies if IRAD = 2 or 3;
= $-S$ to choose mesh appropriate for cases with total $2S+1$, followed by data as for IMESH = 2.
Note, for neutral targets, IMESH = 0 is the only valid option.

INAST : the number of $(N+1)$-electron states. In STGH setting INAST=0 forces the program to loop over all available symmetries (normal operation).

INDATA {10} [STG1 NAMELIST] :
optional parameter for the 'ORBITAL.DAT' input unit number to read user-supplied orbital function input data (either CIV3 or S.S.), i.e. starting with the KEY = −9 header record in the case of S.S. type input (either a blank record or KEY = −5 in position T1,I5 can serve as a terminator). Reset internally.

IOUT=ITAPE3 {3} [STG1 NAMELIST] :
optional parameter for the unformatted 'STG1.DAT' output unit number from STG1. Reset internally.

IOUTDA=JDISC1 {21} [STG1 NAMELIST] :
optional parameter for the direct access 'RK.DAT' output unit number. Reset internally.

IOPT1 [STG4] :

$= -L$ for all symmetries up to L (or $2J$), followed by top-up in omega;
= 1 for all symmetries available;
= 2 for selected $(N+1)$-electron symmetries.
IP [STG4] : = 0 for even parity, = 1 for odd parity, of selected symmetry.
IP1,IP2 [STG4] : = 0 for even parity, = 1 for odd parity, of initial, final state.
IPERT {0} [STG4] :
= 0 to switch off long-range coupling, > 0 to include coupling:
= 1 no top-up;
= 2 top-up, i.e. include estimate for higher partial waves in total collision strength.
IPHOT {0} [RECUPD] : NOT USED.
IPOLPH :
= 0 for target calculations only (i.e. no scattering);
= 1 for electron scattering only (i.e. no dipole matrices);
= 2 for photoionization and/or electron scattering (normal option).
IPOT(NPOT) [STG1] : defines the zero-order potential energy function in Eq. (79).
IPRINT {0} : debug parameter. See individual program modules for descriptions.
IPSEUD {0} [STG1] :
= 1 if model potentials for each l are to be read from ITAPE1;
$= \pm 1$ if model potentials for each l are to be read in S.S. type input mode, where they are read as effective charges $\mathcal{Z}_l(r)$. As explained in the description of routine SPNORB the option -1 allows computation of the effective spin-orbit parameters ζ_l from the derivative of the pseudo-potential — contrary to normal $+1$ when the Blume-Watson formalism is applied, which accounts correctly for exchange contributions from closed shells.
IPTY(NAST) : = 0 for even parity, = 1 for odd parity, of each target state.
IPUNCH {0} : not normally used.
Sometimes used as a switch to activate a program section for output. See individual program module descriptions for details.
IRAD(NCO) [STG1] : powers of r in orbitals as in Eq. (70).
IRAD [STG4] :
= 0 for electron collisions (produces file XOMEGA);
= 1 for electron collisions and photoionization cross sections (produces files XOMEGA and XSECTN);
= 2 for photoionization (produces file XSECTN);
= 3 for polarizabilities (produces file XBOUND);
= 4 for bound state data (produces file XBOUND).
IRDEC [STG4] : NOT USED.
IRELOP(1) :
= 0 to exclude the mass-correction term in the Hamiltonian;
= 1 to include the mass-correction term in the Hamiltonian.
IRELOP(2) :
= 0 to exclude the one-electron Darwin term in the Hamiltonian;
= 1 to include the one-electron Darwin term in the Hamiltonian.
IRELOP(3) :
= 0 to exclude the spin-orbit term in the Hamiltonian;
= 1 to include the spin-orbit term in the Hamiltonian.
IRK1 : total number of bound-bound two-electron integrals stored in array RKSTO1.
IRK2 : total number of bound-continuum two-electron integrals, or the number of continuum-continuum two-electron integrals for a given (l,l') combination, stored in array RKSTO2.
IRK3 : number of λ values stored in array ISTBC1.

IRK4 : number of λ values stored in array ISTBB1.
IRK5 : number of bound-bound one-electron integrals stored in array ONEST1.
IRK6 : number of bound-continuum one-electron integrals stored in array ONEST2.
IRK7 : number of continuum-continuum one-electron integrals stored in array ONEST3 for each continuum angular momentum $l+1$.
IRK8 : number of multipole integrals stored in array RKSTO2.
IRK9 : number of bound-bound Darwin integrals stored in array RDAR1.
IRK10 : number of bound-continuum Darwin integrals stored in array RDAR2.
IRX(I); $I=1$,NIX [STG1] :
the cumulative number of integration steps from the origin to the end of the Ith interval; where the number of points in the Ith interval is N_I = IRX(I) − IRX($I-1$) with N_1 = IRX(1).
IS,IL,IP = ($2S+1$, L, parity), parity, = 0 for even, = 1 for odd [STG4] :
($N+1$)-electron symmetry.
In intermediate-coupling, set (IS,IL,IP) = (0,$2J$,parity).
IS1,IS2 [STG4] : $2S+1$ for initial, final bound state (= 0 in intermediate-coupling).
ISAT(NAST) : $2S+1$, where S is the spin of each target state (= 0 in intermediate-coupling).
ISMIT($l+1$) {0} [STG1 NAMELIST] :
an optional array which declares some of the S.S. type orbitals as correlation orbitals, for angular momentum l. It is the NAMELIST equivalent to MAXNHF − MAXNLG. Thus {ISMIT(1)=40, ISMIT(2)=41, ISMIT(3)=42} singles out the 3 orbitals 4s, 4p, and 4d, and the continuum orbitals will be Schmidt orthogonalised to them.
ISPN,ILRGL,IPTY : the initial ($2S+1$, L or $2J$, parity) values for an ($N+1$)-electron transition on the D00 file. $2S+1=0$ in intermediate-coupling.
IST1($l+1$); $l+1=1$,LRANG1 :
the position in the ONEST1 array where the first bound-bound one-electron integral corresponding to angular momentum l is stored.
IST2($l+1$); $l+1=1$,LRANG1 :
the position in the ONEST2 array where the first bound-continuum one-electron integral corresponding to angular momentum l is stored.
ISTBB1(IRK4) : for the bound-bound two-electron radial integrals, the k values.
ISTBB2(IRK4) : for the bound-bound two-electron radial integrals, the positions in the RKSTO1 array where the first integral corresponding to the ($l_1l_2l_3l_4k$) combination is stored.
ISTBC1(IRK3) : for the bound-continuum two-electron radial integrals, the k values.
ISTBC2(IRK3) : for the bound-continuum two-electron radial integrals, the positions in the RKSTO1 array where the first integral corresponding to the ($l_1l_2l_3l_4k$) combination is stored.
ITAPE1,ITAPE2,ITAPE3,ITAPE4 {1,2,3,4} : intermediate files for transferring data between program modules. See module descriptions.
In module RECUPD, ITAPE1 is used as a switch to decide whether or not to process dipole matrices.
ITARG(i) = $n_i * 1000$ + L2P(i); $i=1$,NCHAN [STG4] :
where n_i is the target state coupled to channel i, and L2P(i) is the channel angular momentum.
ITEMS= (NOPEN $*$ (NOPEN + 1))/2 [STG4] :
number of elements in the upper triangle of the (symmetric) K-matrix .
ITMP(MTC) = $100*i+f$ [RECUPD] :
identifies initial i and final f states for MTC term-coupling coefficients for given angular momentum J on IPUNCH.
ITOTAL {999} : the total number of data blocks required on ITAPE3 (normally set to a large value).
In STG1 each block of integrals is counted.
In STG2 each ($N+1$)-electron symmetry gives rise to four blocks of data on the output file: continuum-continuum, continuum-bound, bound-bound Hamiltonian matrix blocks, together with the associated asymp-

totic coefficients.
IWRITE {6} : standard output.
IZESP {0} [STG1] : not used for a non-relativistic calculation.
= 0 normally, for inclusion of screening by closed shell electrons in evaluation of the spin-orbit interaction;
$= -1$ to exclude screening in the spin-orbit interaction;
> 0 (e.g. LRANG1) if screening factors (ZESP(L), $L = 1$, IZESP) will be supplied for use in routine SPNORB.
J1QNRD(i, k, j); $i = 1, 2 *$ NOCCSH(j) $- 1$; $k = 1, 3$; $j = 1$, NCFG :
angular momentum quantum numbers for the jth configuration: $k = 1$ is seniority; $k = 2$ is $(2l + 1)$; $k = 3$ is $(2s + 1)$. $i = 1$, NOCCSH(j) is the set of quantum numbers for the ith shell; the remaining NOCCSH(j) $- 1$ are quantum numbers resulting from the coupling between the shells (see Hibbert [50,51]).
J2, JP = ($2J$,parity), where J is the total angular momentum of the $(N + 1)$-electron system.
JBBPOL(LRANG1,LRANG2) : same as IBCPOL, but for the bound-Buttle dipole integrals.
JBCPOL(LRANG2,LRANG2) : same as ICCPOL, but for the continuum-Buttle dipole integrals.
JDISC1 {12} [STG1] : direct access 'RK.DAT' output.
JDISC2 {0} : NOT USED.
JJ, JPTY(IJNAST) = ($2J$,parity) for each target level.
JLRGL : the final L or $2J$ value for an $(N + 1)$-electron transition on the D file.
JNAST : the total number of N-electron atomic or ionic target J levels.
JNTCON(i); $i = 1$, JNAST :
the number of states from STG2 which couple to the ith target level.
JP(IJNAST) : the parity of the $(N + 1)$-electron symmetry: 0 = even, 1 = odd.
JPTY(JNAST) : the parity of each target level: 0 = even, 1 = odd.
JRELOP(1) :
= 0 to exclude the mass-correction term in the Hamiltonian;
= 1 to include the mass-correction term in the Hamiltonian.
JRELOP(2) :
= 0 to exclude the one-electron Darwin term in the Hamiltonian;
= 1 to include the one-electron Darwin term in the Hamiltonian.
JRELOP(3) :
= 0 to exclude the spin-orbit term in the Hamiltonian;
= 1 to include the spin-orbit term in the Hamiltonian;
$= -1$ in module STG2, for calculation of term-coupling coefficients in RECUPD.
JRK2 : the number of continuum-continuum two-electron integrals for a given (l, l') combination, stored in array RKST02. Set negative if there are more integrals for that particular (l, l').
JRK8 : the total number of multipole integrals stored in the SKST02 array.
JSPN, JLRGL, JPTY : the final ($2S + 1$, L or $2J$, parity) values for an $(N + 1)$-electron transition on the D00 file. $2S + 1 = 0$ in intermediate-coupling.
JSYM : number of unique N-electron target symmetries.
KEY [STG1] : S.S. input parameter $= -9, -8, -7, -6$ to distinguish blocks of data.
KOUNT : number of dipole allowed transitions on D file.
KRELOP= $4 *$ IRELOP(1) $+ 2 *$ IRELOP(2) $+$ IRELOP(3) ; {0} [STG1 NAMELIST] :
defines the Breit-Pauli options, which are zero by default unless one specifies KRELOP. Hence KRELOP = 7 switches on all three Breit-Pauli one-electron terms, and includes screening of the spin-orbit interaction (IZESP = 0). KRELOP has two further options:
< 0 for all three one-electron terms, but in this case screening is user-specified: IZESP =| KRELOP |;
= 9 for all three one-electron terms, but with no screening (IZESP $= -1$).
L, LP (as an ordered pair on the STG1 output file ITAPE3) :
the continuum l value on each side of a continuum-continuum two-electron integral.

L2P(i); $i = 1$, NCHAN : orbital angular momentum of continuum electron in ith channel.

LAM {3} :

= 1 for electron scattering, when the dipole matrix is not needed there is some saving by just evaluating and storing the bound-bound multipole integrals;

= 3 for evaluating all bound-bound, bound-continuum and continuum-continuum dipole integrals; these are required for photoionization calculations.

LAMAX : is the maximum order of multipole integrals to be evaluated. These integrals are required for the long-range potential coefficients and for the dipole matrix (set LAMAX ≥ 2 for photoionization calculations).

LAT(NAST) : the orbital angular momentum of each target state ($2J$ in intermediate-coupling).

LCB {0} [STG1] :

an untested feature, LCB > 0 if some of the continuum basis orbitals are to be treated as part of the bound basis; LCB $- 1$ is the highest angular momentum for such treatment.

LENGTH(n) = NTCON(n) $*$ (NTCON(n) $+ 1)/2$:

number of elements in upper triangle of N-electron Hamiltonian matrix for given symmetry on STG2 output file ITAPE3, for processing in module RECUPD.

LJCOMP(MAXORB) : the l value of each bound orbital.

LL, LSPN, LPTY = (L,$2S + 1$,parity) [STG2] of each target state.

LPTY : = 0 for even parity, = 1 for odd parity, of target state.

LRANG1 : specifies inclusion of bound orbitals with $l + 1 \leq$ LRANG1.

LRANG2 : specifies inclusion of continuum orbitals with $l + 1 \leq$ LRANG2.

LRGL, NSPN, NPTY = (L,$2S + 1$,parity): for $(N + 1)$-electron symmetry.

In intermediate-coupling, set (LRGL, NSPN, NPTY) = ($2J$,0,parity).

LRGL : the total orbital angular momentum L, or $2J$ in intermediate-coupling.

LSPN : = $(2S + 1)$, where S is the total spin of target state, = 0 in intermediate-coupling.

LSVALU(JNAST, k) : defines the position in the LAT and ISAT arrays (read on unit number ITAPE2 from STG2) of the 'parent' L and S values of the kth configuration which is coupled to each of the JNAST target levels.

MAG [STG4] : magnetic quantum number.

MAXC {&NR2} [STG1 NAMELIST] :

optional parameter to override the effective NRANG2 computed from MAXE; a very large value for MAXC (e.g. 99) for the number of continuum functions reduces to the array length &NR2.

MAXE [STG1 NAMELIST] :

compulsory parameter to specify the highest collision energy in Ry, ensures that S.S. type orbital input is suitably interpolated to secure enough tabulation points. And unless MAXC is specified it assigns a value to NRANG2 such that $k^2_{NRANG2} \approx 2 *$ MAXE is satisfied.

MAXLS [STG1 NAMELIST] :

compulsory parameter to specify the largest total target orbital angular momentum value L.

MAXM1 : the number of different L values occurring in the array CGC.

MAXNHF($l + 1$); $l + 1 = 1$, LRANG1 :

the maximum principal quantum number n of the bound orbitals for angular momentum l.

MAXNLG($l + 1$); $l + 1 = 1$, LRANG1 :

the maximum principal quantum number of those bound orbitals to which the continuum orbitals are to be Lagrange orthogonalised, for angular momentum l. The continuum orbitals are then Schmidt orthogonalised to any remaining bound orbitals (e.g. the pseudo-orbitals). If no Lagrange orthogonalisation is required for a given angular momentum then set MAXNLG($l + 1$) = l.

MAXNC(LRANG1) :

the maximum principal quantum number n of the core electrons to be represented by a model potential.

MAXORB : the total number of bound shells.

In STG2 input, the user can supply −MAXORB for the program to automatically generate MAXORB (n,l) values, so the record containing the NJCOMP,LJCOMP arrays does not have to be input.

MAXPW [STG1 NAMELIST] :
compulsory parameter to specify the largest $(N+1)$-electron L.

MCFGP : the number of $(N+1)$-electron configurations in the initial state.

MCHAN : the number of channels in the initial state.

MNAL,MXAL(MAXORB) [STG2] : the minimum, maximum number of electrons required in each shell.

MNP1 : = NRANG2 $*$ NCHAN + NCFGP, i.e. order of Hamiltonian matrix.

MNP2 : = NOTERM $*$ NCHAN + NCFGP, i.e. order of Hamiltonian matrix.

MORE :
= 0 if no more $(N+1)$-electron symmetries;
= 1 if further $(N+1)$-electron symmetries.

MPTY [STG4]= ±IS, where the sign is determined by parity, −1 for odd parity, +1 for even parity.

MTC [RECUPD] : number of term-coupling coefficients for given angular momentum J on IPUNCH.

MXAL(MAXORB) : the maximum number of electrons required in each shell.

MXE,E0,EINCR [STG4] : number of energies at equal intervals, initial energy, increment.

NAST : the number of N-electron states.
In STG2, if JRELOP(3) $\neq$ 0, NAST = number of N-electron *configurations*

NBUG1... 9 {0} : debug parameters. See module descriptions.

NBUT {1} [STGH] : NOT USED.

NCFG : the total number of different configurations for all N-electron states.

NCFGP : number of $(N+1)$-electron configurations for given symmetry.

NCHAN : total number of coupled channels for given symmetry.

NCO [STG1] : number of Slater-type coefficients in Eq. (70).

NCONAT(NAST) : number of channels coupled to each N-electron state.

NCONHP= NRANG2 $*$ NCHAN, i.e. total number of continuum terms.

NCUT {0} [STG2] : is the total number of configurations generated or read if not all of them are to be stored. NCUT applies to the target states and allows you to exclude configurations from the target description. This facility is only for the experienced user. See array IKIP.

NDIAG {1} [STG2] :
= 0 to read the N-electron configuration coefficients and energies;
= 1 to diagonalize the N-electron Hamiltonian to produce the configuration coefficients and energies (normal option).

NELC = N : the number of electrons in the N-electron target.

NELCSH(i,NCFG) :
the number of electrons in the ith occupied shell for each N-electron configuration.

NELCOR [STG1] : is set to NELC.

NEN [STG4] : number of impact energies.

NEST {0} [STGH] :
normally zero, but can be specified as the number of target states, whose energies are then read in free format on the next record or records. See Section 5.4.

NIX [STG1] :
(≥ 2) is the number of intervals covering the range $0 < r <$ RA, typically NIX ≈ 5.

NJCOMP,LJCOMP(MAXORB) : the (n,l) value of each bound orbital.

NKEY [STG2] : set $\neq$ 2 if the N-electron configurations are to be automatically generated:
= −1 for minimum data;
= 0 for specifying the same criterion for each state (normal option);
= 1 for different criteria for each state, records 8-11 are therefore repeated for each state;

= 2 if the N-electron configurations are to be read in explicitly; and all configurations for all states are read as in from the formatted input file JREAD.

NN {1} [STG4] : = 1, since only one bound state for each symmetry is considered.

NOCCSH(NCFG) : the number of occupied shells in each N-electron configuration.

NOCORB(i,NCFG) : the position of the ith occupied shell in the NJCOMP and LJCOMP arrays for each configuration.

NOPEN [STG4] : number of open channels.

NOPTN {0} [STG2] :
= number of basic configurations specified in record 10 to restrict the number of electron excitations;
= −1 for minimum input data;
= −2 for no configurations to be generated for this state (this facility is only used if NKEY = 1 and there are more than one N-electron states with the same symmetry; then the configurations need only be generated once).

NOT1,NOT2 {NRANG2} [STGH] :
as in the old STG3 one can still specify an interval NOT1 $\leq$ NOT2 $\leq$ NRANG2 for a convergence test, but the answers no longer make sense in physics, because they are all Buttle corrected for NRANG2 continuum functions.

NOTERM= NRANG2 normally : is the number of continuum basis orbitals used for each of the channel angular momenta.

NPOT {0} [STG1] :
= 0 normally, in this case the zero-order potential in Eq. (15) is automatically generated;
$>$ 0 to read in the zero-order potential function as an analytic function — as in Eq. (79). NPOT is the number of terms;
$<$ 0 can be used along with S.S. type numerical input. See the description of routine POTF.

NPTS [STG1] : number of radial mesh points in (0,RA).

NPTY : parity, = 0 for even, = 1 for odd.

NRANG2 : is the number of continuum basis orbitals to be evaluated for each of the LRANG2 angular momenta.

NS,LS – as an ordered pair – [STG1] : (n,l) quantum numbers of orbital on S.S. file.

NSHELL,LSHELL – [STG1] : (n,l) quantum numbers of each shell in order of occupancy, in the ground state configuration. Option for static potential determination in routine POTF.

NSP : N-electron Hamiltonian matrix for given $LS\pi$ symmetry.

NSPN : = $(2S+1)$, where S is the total spin, or = 0 in intermediate-coupling.

NTCO :
= NTCON if JRELOP(3) = 1;
= −(number of target terms of given symmetry) if JRELOP(3) = −1.

NTCON(NAST) : the number of configurations stored for each target state.

NTRAN [STG4] : number of energetically allowed transitions in a triangle of the (symmetric) collision strength matrix, the diagonal is included if the target is neutral, otherwise not.

NTYP(NAST,j); j = 1,NTCON :
the position of the jth configuration (in the order generated by the CONFIG package or read in from JREAD) for each target state.

NXCITE(m); m = 1,NOPTN [STG2] :
the maximum number of electrons to be excited from the mth basic configuration.

NZ = Z : the atomic number of the atom or ion ($Z \geq N \geq 1$).

NZED = Z : the atomic number of the atom or ion.

OMEGA(NTRAN) [STG4] : total collision strength for each transition (i.e. summed over all partial waves, with top-up if IPERT = 2).

OMEGA2 [STG4] : frequency of the radiation in Ry.

ONEST1(IRK5) : the array containing the bound-bound one-electron integrals $Q(m,n)$ as in Eq. (78). The lower triangle of the matrix of elements defined by (m,n) are stored consecutively by rows.

ONEST2(IRK6) : the array containing the bound-continuum one-electron integrals. The matrix of elements defined by (m,n) are stored consecutively by rows.

ONEST3(IRK7,LRANG2) : the array containing the continuum-continuum one-electron integrals. The lower triangle of the matrix of elements defined by (m,n) are stored consecutively by rows.

PHSUM [STG4] : eigenphase sum (modulo π).

PX(NPTS) [STG1] : contains model potential.

QNMAX [STG4] : largest allowed value of effective n.

RA : the R-matrix boundary radius in a_o.

RDAR1(IRK9) : bound-bound Darwin integral $I_{D_1}(i,j)$ as in Eq. (83).

RDAR2(IRK10) : bound-continuum Darwin integral.

RDAR3(IRK7) : continuum-continuum Darwin integral.

RK(NOPEN,NOPEN) [STG4] : K-matrix element.

RKST01(IRK1) : two-electron integrals $R^k(n_1l_1,n_2l_2,n_3l_3,n_4l_4)$ as in Eq. (73).

RKST02(IRK2) :

the array containing all multipole integrals, and two-electron radial integrals involving continuum orbitals.

Multipole integrals RKST02 contains all bound-bound, bound-continuum and continuum-continuum multipole length $I_L^k(nl,n'l')$ and dipole velocity $I_V(nl,n'l')$ integrals, as in Eq. (82) and Eq. (69). When $k = 1$, two numbers are stored consecutively, the first is the length dipole integral I_L^1 and the second is the velocity dipole integral I_V. When $k > 1$ only the length multipole integral I_L^k is stored. For each l,l',k, the n,n' values are stored as follows:

when $l = l'$, the lower triangle of the bound-bound and continuum-continuum are stored consecutively by rows (in pairs if $k = 1$);

when $l \neq l'$, the elements are stored consecutively by rows (in pairs if $k = 1$).

Bound-continuum two-electron integrals $R^k(n_1l_1,n_2l_2,n_3l_3,n_4l_4)$ in Eq. (73), where (n_4l_4) is the continuum orbital. The integrals defined by $(n_1n_2n_3n_4)$ are stored in RKST02 consecutively by rows, i.e. n_4 varies most rapidly, n_3 varies next most rapidly, and so on over their allowed ranges.

Continuum-continuum two-electron integrals $R^k(n_1l_1,n_2l_2,n_3l_3,n_4l_4)$ in Eq. (73), where (n_2l_2), (n_4l_4) are continuum orbitals in the direct integral, and (n_2l_2), (n_3l_3) are continuum orbitals in the exchange integral.

RMASS1(IRK5) : bound-bound mass-correction $I_{\text{mass}}(i,j)$ integrals, Eq. (84).

RMASS2(IRK6) : bound-continuum mass-correction integrals.

RMASS3(IRK7,$l+1$) : continuum-continuum mass-correction integrals for $l = 1$,LRANG2.

RONE {1.0} [STG4] : NOT USED.

RSPOR1(IRK5) : bound-bound spin-orbit $I_{SO}(i,j)$ integrals, Eq. (86).

RSPOR2(IRK6) : bound-continuum spin-orbit integrals.

RSPOR3(IRK7,$l+1$) : continuum-continuum spin-orbit integrals for $l = 1$,LRANG2.

SIGL,SIGV(NAST) [STG4] : photoionization cross section (in Mb) – in length and velocity forms – to each final target state.

SKST02(JRK8) : dipole radial integrals involving Buttle term.

TEMP(MTC) [RECUPD] :

MTC term-coupling coefficients for given angular momentum J on IPUNCH.

TEXT [STG1] : S.S. input parameter, contains text, probably not used.

TITLE(18)*4 :

contains 72 characters of text and is printed out on unit IWRITE as a heading for the calculation.

In STG1, if the first four characters are:

S.S. then radial orbital input is read in the format described by Crees et al. [38] and generated by routine

RADIAL of the SUPERSTRUCTURE code [43];
CIV3 then radial orbital data in CIV3 format [52] is expected;
STO- then radial orbital data in STG1 format is expected.
In each of these cases the NAMELIST format is invoked.

`TR,TI(NOPEN,NOPEN)` [STG4] : T-matrix element - real and imaginary parts.

`UJ(NPTS,`j`)` [STG1] :
tabulation of all bound $P_{nl}(r)$ and continuum $u_{ij}(r)$ orbitals. The first j = 1,`NBOUND` locations are bound, and are defined by the user input.

`VALUE(`k`)` $= E_k$; k = 1,`MNP2` :
$(N+1)$-electron Hamiltonian eigenvalues stored on H file, as in Eq. (13) and Eq. (24).

`W1` [STG4] : photon energy in Ry.

`WMAT(`i,k`)` $\equiv w_{ik}$(RA); i = 1,`NCHAN`; k = 1,`MNP2` :
surface amplitudes stored on H file, as in Eq. (20).

`X(NTC)` : eigenvector of the N-electron Hamiltonian matrix.

`XL,XV` [STG4] : total photoionization cross section - in length and velocity forms - in Megabarns (i.e. summed over all final states included).

`XPART(`i,j`)` [STG4] : partial collision strength for transition between target states i and j.

`XPOT(NPOT)` [STG1] : the zero-order potential energy function in Eq. (79).

`XR(NPTS)` [STG1] : values of radial mesh points r.

`ZE(NCO)` [STG1] : bound orbital exponents as in Eq. (70).

`ZESP(`$l+1$`)`; $l+1$ = 1,`IZESP` {not normally specified} [STG1] :
= the screening factor (i.e. a number between 0 and 1) for the spin-orbit parameter to angular momentum l (Blume and Watson [18]): effective $\zeta'_{nl,kl}$ = ZESP$(l+1) * \zeta^0_{nl,kl}$, where k stands for bound or continuum orbitals.

10. Additional programs to interface with current codes

In this section we summarise the computer programs which are available to interface with the current R-matrix codes. These either provide input data, or process the output data, or augment the current codes, as outlined below. All programs are written in FORTRAN, and most are available from the CPC program library.

10.1. Programs providing input data

These programs can provide input data for use in module STG1 of the R-matrix package.

- CIV3: a general program to calculate configuration interaction wavefunctions and electric-dipole oscillator strengths, Hibbert [52]. More up-to-date versions are available for fine-structure transitions, and for polarized pseudo-state generation (CIVPOL) (Hibbert, private communication).
- SUPERSTRUCTURE: atomic structure package, Eissner et al. [43].
- MCHF: a general multi-configurational Hartree-Fock program, Fischer [46] (an interface to STG1 is under construction).

10.2. Programs to process the output data

These programs can normally use reactance $\boldsymbol{K}$- or $\boldsymbol{T}$-matrices output from external region codes such as STG4.

- Fine structure collision strengths from LS-coupled reactance matrices:
  JAJOM: is widely used, more up-to-date versions are available from the Iron Project, Saraph [69,70];

LSTOIC: has not been used with R-matrix programs, Clark [33].

- Resonance fitting:
  RESON: detection and fitting of Breit-Wigner resonances, Tennyson and Noble [82];
  RESFIT: multichannel resonance fitting, Bartschat and Burke [7].
- Angular distributions etc. :
  SCATTERING AMPLITUDES: amplitudes for scattering of electrons by hydrogenic and alkali-like systems, Moores [59];
  SCATTAMPREL: amplitudes for scattering of electrons by atomic systems including relativistic effects, Bartschat and Scott [8];
  OBSERVABLES: observable quantities from scattering amplitudes for inelastic electron-atom collisions, Bartschat [6].
  MOMTRANF: differential and total cross sections for electron-atom or ion scattering using the momentum transfer formalism, Salvini [68];
  DCS2: differential and integral cross sections for quantum mechanical scattering problems from reactance or transition matrices, Onda et al. [63].
- Belfast Atomic Data Bank: Hughes et al. [53] and Berrington et al. [15].

10.3. Alternative external region codes

With suitable interfacing, these programs can take the internal region information in the H and D files output from module STGH, and provide the external region solutions to complete the problem i.e. they can replace module STG4.

- Opacity Project external region codes, STGF, STGB, STGBB, STGBF: Seaton (private communication), Berrington et al. [11]. Restricted to ionic targets:
  STGF incorporates efficient Coulomb routines and a perturbation technique for the electron scattering problem;
  STGB calculates bound states;
  STGBB and STGBF allow bound-bound and bound-free radiative data for ground and excited states to be calculated.
- FARM: a Flexible Asymptotic R-Matrix code, Burke and Noble [28]. This program incorporates the R-matrix propagator techniques of Baluja et al. [3] and Light and Walker [58], and the accelerated asymptotic expansion method of Noble and Nesbet [60], to provide an efficient program for solving the coupled equations for both neutral and ionic targets. Restricted to electron excitation.
- VPM: Croskery et al. [39]. Uses a variable-phase method to solve the coupled equations for electron scattering. It has also been used in calculations of atomic free-free transitions. Restricted to neutral targets.
- Norcross' program [61], used in the original versions of the R-matrix programs, has now been superseded by Crees [37].
- Coulomb programs: Barnett et al. [4] for open channel energies; Bell and Scott [9] for closed channel energies. These have been used for electron-ion scattering cases where the external region equations can be completely uncoupled, and the solutions are just Coulomb functions evaluated on the R-matrix boundary. Restricted to ionic targets.

10.4. Other programs to augment current codes

- No-exchange R-matrix program: Burke et al. [29]. This program is for electron-atom and -ion scattering in LS-coupling, and can be used for cases where exchange between the scattering electron and the target electrons can be ignored, e.g. for higher partial waves.

- RMATRX2: Burke et al. [26]. A new *R*-matrix approach for calculating cross sections for electron-impact excitation of complex atoms and ions. This new approach, based on an expansion of the total wavefunction in target configurations rather than in individual target states, and taking advantage of the special status of the scattered electron in the collisional wavefunction, enables the angular integrals to be performed very much more efficiently than hitherto. It also enables electron correlation effects in the target and in the electron-target collision complex to be treated consistently, eliminating pseudo-resonances. A major new program package has been written that implements this approach for electron collisions. Photoionization has not yet been implemented, nor have relativistic effects.
- Dirac Atomic *R*-matrix Codes (DARC): Norrington and Grant [62]. A general program package based on GRASP [41] for electron-atom and electron-ion collisions has been written. Use of the Dirac Hamiltonian is essential when relativistic effects are dominant for high Z targets.

11. Test runs

Two test runs will be described here, associated with the two mutually-exclusive modes of operation of the programs, namely an *LS*-coupling run, and a Breit-Pauli run.

In order to maintain consistency with previous *R*-matrix program descriptions (Berrington et al. [12,13]; Scott and Taylor [73]), and to allow comparisons with the fully relativistic treatments of Chang [31] and Norrington and Grant [62], we examine the similar electron scattering and photoionization processes:

$$e^- + \mathrm{Ne}^+ \to e^- + \mathrm{Ne}^+$$

$$h\nu + \mathrm{Ne} \to e^- + \mathrm{Ne}^+$$

The ionic states of Ne^+ are represented by single configurations: $1s^2 2s^2 2p^5$ and $1s^2 2s 2p^6$, in terms of the same bound 1s, 2s and 2p orbitals, which are the analytic Hartree-Fock orbitals given by Clementi and Roetti [34].

In a more realistic but longer calculation, additional pseudo-orbitals, optimised in programs CIV3 or SUPERSTRUCTURE, could be introduced to obtain improved configuration-interaction wavefunctions for the Ne^+ ionic states. Moreover, to obtain converged electron scattering cross sections, many more partial waves may be needed than the two considered in these test runs.

The test runs are therefore rather restrictive in character, and do not fully illustrate the versatility of the program; nevertheless they do serve both as a useful benchmark to ensure that the programs are running correctly, and as a template for users to construct their own runs. In the latter case, the user should review the use of the debug parameters, as these could produce large amounts of unnecessary output in bigger runs.

11.1. LS-coupling test run

- The electron scattering test corresponds to elastic and inelastic scattering of electrons by the two lowest terms of the ion Ne^+:

$$\mathrm{Ne}^+(1s^2 2s^2 2p^5)\,^2P^o$$

$$\mathrm{Ne}^+(1s^2 2s 2p^6)\,^2S^e$$

  where the total system of electron plus Ne^+ ion is in the $^1S^e$ or $^1P^o$ state.
- The photoionization process corresponds to photoionization from the ground state of Ne leaving the Ne^+ ion in one of these two ionic states,

$$h\nu + \mathrm{Ne}(1s^2 2s^2 2p^6)\,^1S^e \to \mathrm{Ne}^+(1s^2 2s^2 2p^5)\,^2P^o + e^-(ks, kd)$$

$$\rightarrow \mathrm{Ne}^+(1s^2 2s 2p^6)\,^2S^e + e^-(kp)$$

where the final state is a $^1P^o$ state.

11.1.1. STG1 test run - LS-coupling

The input data is shown below.

```
STG1 TEST CASE :  E + NE II (1S,2S,2P) non-relativistic
0    0    0    0    0    0    0    0    0    0
0    0    0    0    1    0    0    1    0
0  999    0    0    0    0    0
9   10    2    3   10    2    3    1    0    0
2    2
2    2
5
1    1    2    2    2
 9.8112100     16.0692000     3.7237700     8.8049500     2.4772400
 0.91057       0.03736        0.00462       0.06561      -0.00102
5
1    1    2    2    2
 9.8112100     16.0692000     3.7237700     8.8049500     2.4772400
-0.23117      -0.00377        0.59087      -0.09319       0.50552
4
2    2    2    2
 2.5517500      4.7006400     1.7533600    10.1572000
 0.58442        0.31344       0.16267       0.01293
 5.0            0.0
```

The test run output from STG1 begins with a printout of the user-supplied data, which is explained in Section 2.3.

The bound orbitals are first read in, and their orthonormality checked. Since they become negligible ($< 4 \times 10^{-3}$) at $r = 5$ au, the boundary RA is chosen to be at this point:

```
R-MATRIX BOUNDARY CONDITIONS
----------------------------

  RA   =    5.00000
  BSTO =     .00000

AMPLITUDE OF THE FUNCTIONS AT RA
--------------------------------
  1s  ORBITAL -1.2E-06
  2s  ORBITAL  5.9E-04
  2p  ORBITAL  3.5E-03
```

The program automatically corrects the bound orbitals using Eq. (68) so that they vanish on the boundary. An integration mesh is generated by the program to give sufficient mesh points, both in the peaks of the bound orbitals and in the oscillation of the continuum orbitals. As discussed in Section 1, the zero-order potential in Eq. (15) is arbitrary, and can be chosen by the user or generated by the program. We chose the latter in the test run by setting NPOT=0, to generate the static central potential of the 9-electron ground state using the bound orbitals supplied.

The continuum orbitals are now generated by solving Eqs.(15-16); there are NRANG2=10 orbitals for each continuum angular momentum up to l = LRANG2−1 = 2. The calculated eigenvalues and their boundary amplitudes are printed out in the test run output. Here is a sample:

```
ORBITALS FOR L = 0
```

```
AMPLITUDE AT RA     EIGENVALUE (RYD)     NODES

       .65426               -.3732          2
     -.73047                 .8180          3
       .69103               3.1730          4
     -.67287                6.4567          5
       .66324              10.6130          6
     -.65738               15.6244          7
       .65345              21.4796          8
     -.65056               28.1712          9
       .64832              35.6916         10
     -.64651               44.0353         11
```

Also shown are the overlap integrals between the bound and continuum orbitals generated for each l; the overlap integrals should form a unit matrix, so this (optional) output gives some indication of the numerical accuracy of the integrations.

All radial integrals associated with this bound and continuum orbital basis are then calculated. With NBUG8=1 in the test run, the bound-bound integrals are printed out, and these are shown in the output. All integrals are written in unformatted form to the output files for use in program STG2.

A useful facility not shown here is a dimension test run with NBUG7=1. It is particularly useful to establish the number of two-electron radial integrals to be stored in memory in both STG1 and STG2, in order to be able to supply a reasonable value for the &MEM preprocessing parameter (see Section 8).

11.1.2. STG2 test run – LS-coupling

The input data is shown below.

```
STG2 TEST CASE :  E + NE II (1S,2S,2P) non-relativistic
0   0   0   0   0   0   0   0   0   0   0   0
0   0   0   0   0   0   2   1   3
0  999  2   0   0   0
3   9   2  -1   0   2  -1   0   1
1   0   2   0   2   1
2   1   5
1   2   1
0   2   0
2   1   5
0   1   0
1   1   1
```

The test run output from STG2 contains a printout of the user-supplied data, which is explained in Section 3.3.

The user specifies the orbital angular momentum, spin and parity of each N-electron target state being included in the calculation; in the test run these are the two Ne^+ states: $^2P^o$ and $^2S^e$. The N-electron configurations are generated in STG2 with the criterion that the minimum number of electrons in each shell should be given by $1s^22s2p^5$. The test run output shows that this criterion produces the required single configurational states: $(1s^22s^22p^5)^2P^o$ and $(1s^22s2p^6)^2S^e$. Since NDIAG=1 here, the energies of these two states, together with the CI coefficients, are determined by diagonalizing the target Hamiltonian as in Eq. (6):

```
NTYP =  1
AIJ =  1.0000000
ENAT =    -127.817513

NTYP =  2
AIJ =  1.0000000
ENAT =    -126.733017
```

The user then specifies the total orbital angular momentum, spin and parity of each $(N+1)$-electron state; here we examine just two: ${}^1S^e$ and ${}^1P^o$ (the former defines the initial state for photoionization). For each total symmetry, the $(N+1)$-electron configurations are generated using the same criterion as above (there is in fact only one such configuration: $1s^22s^22p^6$); the channel coupling arrays are generated; the $(N+1)$-electron Hamiltonian matrix is constructed as in Eq. (14): and the long-range potential coefficients calculated as in Eq. (29). These data are written to the unformatted output file (ITAPE3) for use in STGH.

Because IPOLPH=2 is specified, STG2 proceeds to calculate dipole matrices as in Eq. (48), (in length and velocity form) involving all $(N+1)$-electron states satisfying the dipole selection rules. The test run output shows the calculation of the dipole matrix for ${}^1S^e$ - ${}^1P^o$. These dipole matrix elements are written out to the second unformatted output file (ITAPE4) for use in STGH.

The unformatted files from STG1 are normally deleted after the STG2 run, as they are no longer needed.

11.1.3. STGH test run – LS-coupling

The input data is shown below.

```
 STGH TEST CASE :  E + NE II (1S,2S,2P) non-relativistic
 0   0   0   0   0   0   0   0   0
 0   0   0   0   0   0   0   1   0
 0   0   2
 1  10  10   1   0   0
 0   1   0
```

The test run output from STGH contains a printout of the user-supplied data, which is explained in Section 5.3.

STGH loops over each total symmetry to diagonalize the $(N+1)$-electron Hamiltonian matrices supplied on input; the test run output shows the resulting eigenvalues for the two symmetries concerned: ${}^1S^e$ and ${}^1P^o$. For the first symmetry

```
ENERGY LEVELS WITH RESPECT TO THE GROUND STATE (AT  -255.63503 RYDBERGS)
    46.08270    42.71760    37.73945    34.70748    30.22081    27.51799
    23.53334    21.15471    17.68543    15.62532    12.68740    10.93153
     8.54977     7.04874     5.28327     3.91398     2.93991     1.79142
     1.46516     -.10568    -1.54720
```

The basic scattering data, including the long-range potential coefficients, and the eigenvalues and surface amplitudes of Eq. (26), are written to the unformatted H file, as described in Section 5.6.

Because IPOLPH=2 is specified, dipole matrices are input and transformed as in Eq. (49) in STGH by multiplying through by the eigenvectors from the $(N+1)$-electron states associated with the transition; in this case for ${}^1S^e$ - ${}^1P^o$. These transformed dipole matrix elements are output to the D files, as described in Section 5.7.

The unformatted files from STG2 are normally deleted after the STGH run, as they are no longer needed. In contrast, the H and D files are normally archived for later use.

11.1.4. STG4 test run – LS-coupling

The input data is shown below.

```
  -3  1  1                  :IPRINT, IRAD, IPERT
   1.E-3                    :AC
   1.                       :RONE
   1                        :IMESH
2 2.5 0.5
   2
1 0 0
1 1 1                       :SLPI CASES
```

The test run output from STG4 contains a printout of the user-supplied data, which is explained in Section 6.3. Note the IRAD=1 option has been chosen here: for the purposes of the test run, both electron scattering and photoionization cross sections are calculated for the same scattering states and electron energy range. The accuracy parameter (AC) has been set to 10^{-3}.

- In the case of electron scattering, the collision strengths for $e^- + Ne^+$ ($1s^22s^22p^5\ {}^2P^o$ – $1s^22s2p^6\ {}^2S^e$), summed over the two partial waves (${}^1S^e$ and ${}^1P^o$) for each energy, 2.5 and 3.0 Ry, are output in file XOMEGA:

```
  10    9    2     OMEGA
  .000000E+00   .216899E+01
 2.50000E+00        1 9.296E-02
 3.00000E+00        1 9.344E-02
```

  with the partial collision strengths in file XDUMP (since IPRINT = −3):

```
NE = 9  NZ =10    2-STATE      RA =   5.0 L2=  3 K-MATRIX
 S L PI   ENERGY    PARTIAL COLLISION STRENGTH
1  0 0 2.500000E+0 1.831E+00 4.109E-02 7.116E-01
1  1 1 2.500000E+0 1.065E+00 5.186E-02 3.024E+00
1  0 0 3.000000E+0 1.798E+00 4.190E-02 5.508E-01
1  1 1 3.000000E+0 8.431E-01 5.154E-02 3.789E+00
```

- In the case of photoionization, the first symmetry specified is taken as the initial state of Ne (i.e. ${}^1S^e$), and the lowest bound state is calculated (this is the ground state of Ne, $1s^22s^22p^6$):

```
                BOUND STATE ENERGY =  -128.5898588 A.U.
SEPARATION FROM THE LOWEST POLE = -.1253460E-02 A.U.

RELATIVE TO IONIZATION THRESHOLD=    -1.54469 RYD.
                      EFFECTIVE N =   .8046
```

  The total photoionization cross section from this state to all possible final states (i.e. ${}^1P^o$ for $e^- + Ne^+$) is then calculated for the two photon energies, and output in file XSECTN in length and velocity forms:

```
  10    9    2      PHOTOIONIZATION
  .000000E+00   .216899E+01
   1    0    0    1
 -.154469E+01    2
 0. PHOTON ENERGY(Ryds), CROSS SECTION (Mb) L,V:
  4.044692E+00 7.319E+00 5.601E+00
  4.544692E+00 6.757E+00 5.126E+00
```

Clearly, STG4 can be repeated using the same H and D files with different options, e.g. energy ranges and IPRINT. Note that the D files are not required if only electron scattering or bound state calculations are being performed.

11.2. Breit-Pauli test run

The electron scattering test corresponds to elastic and inelastic scattering of electrons by the three lowest fine-structure levels of the ion Ne^+:

$$Ne^+(1s^22s^22p^5)\,{}^2P^o_{3/2}$$
$$Ne^+(1s^22s^22p^5)\,{}^2P^o_{1/2}$$
$$Ne^+(1s^22s2p^6)\,{}^2S^e_{1/2}$$

where the total system of electron plus Ne$^+$ ion is in the 0^e or 1^o state. The photoionization process corresponds to photoionization from the ground state of Ne leaving the Ne$^+$ ion in one of these three ionic levels,

$$\begin{aligned} h\nu + \mathrm{Ne}(1s^22s^22p^6)\,^1S^e_0 &\rightarrow \mathrm{Ne}^+(1s^22s^22p^5)\,^2P^o_{3/2} + e^-(ks, kd) \\ &\rightarrow \mathrm{Ne}^+(1s^22s^22p^5)\,^2P^o_{1/2} + e^-(kd, ks, kd) \\ &\rightarrow \mathrm{Ne}^+(1s^22s2p^6)\,^2S^e_{1/2} + e^-(kp) \end{aligned}$$

where the final state is a 1^o state.

11.2.1. STG1 run – Breit-Pauli

The input data is shown below.

```
STG1 TEST CASE :  E + NE II (1S,2S,2P) Breit-Pauli
0   0    0    0    0    0    0    0    0    0
0   0    0    0    0    0    0    1    0
0  999   0    1    1    1    0
9   10   2    3   10    2    3    1    0    0
2   2
2   2
5
1   1    2    2    2
 9.81121      16.06920      3.72377       8.80495       2.47724
 0.91057       0.03736      0.00462       0.06561      -0.00102
5
1   1    2    2    2
 9.81121      16.06920      3.72377       8.80495       2.47724
-0.23117      -0.00377      0.59087      -0.09319       0.50552
4
2   2    2    2
 2.55175       4.70064      1.75336      10.15720
 0.58442       0.31344      0.16267       0.01293
 5.0           0.0
```

The test run is similar to the *LS*-coupling case (Section 11.1.1), except that the Breit-Pauli operators in Eqs.(60–62) have been switched on: `IRELOP(1)=1` for the mass-correction term; `IRELOP(2)=1` for the Darwin term; `IRELOP(3)=1` for the spin-orbit term. Extra one-electron integrals are therefore calculated and stored on the output file.

11.2.2. STG2 run – Breit-Pauli

The input data is shown below.

```
STG2 TEST CASE :  E + NE II (1S,2S,2P) Breit-Pauli
0   0    0    0    0    0    0    0    0    0    0    0
0   0    0    0    0    0    2    0    0
0  999   2    1    1    1
3   9    2   -1    0    5   -1    0    1
1   0    2    0    2    1
2   1    5
1   2    1
0   2    0
2   1    5
0   1    0
1   3    0
1   1    1
```

```
1   3   1
2   3   1
```

The test run is similar to the *LS*-coupling case (Section 11.1.2). However, because `IRELOP(3)=1` is specified, for the spin-orbit interaction, the user must specify the orbital angular momentum, spin and parity of each *N*-electron **configuration** being included in the calculation; in the test run there are two Ne$^+$ configurations: $(1s^22s^22p^5)^2P^o$ and $(1s^22s2p^6)^2S^e$. Since `NDIAG=1` here, the target Hamiltonian matrix elements involving these configurations are calculated in STG2 for later diagonalization in RECUPD.

In the Breit-Pauli mode, STG2 calculates the non-fine-structure $(N+1)$-electron Hamiltonian matrix elements required by the right hand sides of Eqs.(104–107), with the *N*-electron configurations as the Φ_i in the first summation of Eq. (12), together with the long-range potential coefficients. In this test run we only require the total symmetry of the system to be 0^e and 1^o (the former defines the initial state for photoionization). Since we must run STG2 for all *LS* symmetries which can contribute to the required $J\pi$ symmetries, this means that we specify five $SL\pi$ symmetries here:

`LRGL=0, NSPN=1, NPTY=0`

$LRGL = 1,\ NSPN = 3,\ NPTY = 0$

$LRGL = 1,\ NSPN = 1,\ NPTY = 1$

$LRGL = 1,\ NSPN = 3,\ NPTY = 1$

$LRGL = 2,\ NSPN = 3,\ NPTY = 1$

(note that `LRGL=0, NSPN=3, NPTY=1` could also couple to 1^o, but there are no coupled channels for this *LS* symmetry). The *LS*-coupled Hamiltonian matrix elements, together with the long-range potential coefficients and other data, are stored on the unformatted output file (`ITAPE3`), for use in RECUPD.

Since we specify `IPOLPH=2`, STG2 then calculates all possible dipole matrices between the $(N+1)$-electron states defined above, subject to the appropriate selection rules, and writes them to the second unformatted output file (`ITAPE4`), for use in RECUPD.

The unformatted files from STG1 are normally deleted after the STG2 run, as they are no longer needed.

11.2.3. RECUPD run – Breit-Pauli

The input data is shown below.

```
RECUPD TEST CASE :  E + NE II (1S,2S,2P) Breit-Pauli
0   0   0   0   1   0   3   0
0   0   0   0   0   0   0   3   0
3   1   0
3   1   1
1   1   0
2
0   0
2   1
```

The test run output from RECUPD contains a printout of the user-supplied data, which is explained in Section 4.4.

The user specifies the $J\pi$ symmetry of each *N*-electron target state to be included in the calculation; in the test run there are three Ne$^+$ levels: $\frac{3}{2}^o$, $\frac{1}{2}^o$ and $\frac{1}{2}^e$. Since `ICHECK=1` here (`NDIAG=1` having been specified in STG2), the energies and CI coefficients of the target levels are calculated by diagonalizing the target Hamiltonian for each $J\pi$ symmetry. These are shown in the test run printout:

```
LEVEL   J   PARITY  ENERGY EI/2RY      (EI-E1)/RY
```

```
  1   1.5    ODD    -127.9569579     0.0000000
  2   0.5    ODD    -127.9532454     0.0074251
  3   0.5   EVEN    -126.8659475     2.1820208
LEVEL:  EIGENVECTOR
  1      1.0000000
  2      1.0000000
  3      1.0000000
LEVEL, LVEC  JNTCON:  COMPONENT TERMS

   1     1     1     1
   2     1     1     1
   3     1     1     2
```

The user then specifies the $J\pi$ symmetries of the $(N+1)$-electron system required. The test run printout shows that the first two LS symmetries contribute to 0^e, while the remaining three contribute to 1^o, the two $J\pi$ symmetries requested.

Each $J\pi$ symmetry is considered in turn, and RECUPD reads from file ITAPE2 the non-fine-structure Hamiltonian matrices calculated in STG2. These are recoupled according to Eqs.(104–107). Then all necessary spin-orbit Hamiltonian matrices are calculated and added to the transformed non-fine-structure Hamiltonian matrices to give the left hand sides of Eqs.(104–107). These transformed Hamiltonian matrices, together with the long-range potential coefficients recoupled according to Eq. (108), are written to the unformatted output file (ITAPE3) for use in STGH.

Since STG2 was run with IPOLPH=2, RECUPD can read from file ITAPE1 the LS-coupled dipole matrix elements in both length and velocity form calculated in STG2. These matrix elements are then combined to form dipole matrices between the specified $J\pi$ states, according to the appropriate selection rules: in this case for $0^e - 1^o$. The transformation is applied as described by Eqs.(112–115). The transformed dipole matrix elements are written to the second unformatted file (ITAPE4) for use in STGH.

The unformatted files from STG2 are normally deleted after the RECUPD run, as they are no longer needed.

11.2.4. STGH run – Breit-Pauli

The input data is shown below.

```
STGH TEST CASE :  E + NE II (1S,2S,2P) Breit-Pauli
0    0    0    0    0    0    0    0    0
0    0    0    0    0    0    0    1    0
0    0    2
1   10   10    1    0    0
0    0    0
```

The test run is similar to the LS-coupling case (Section 11.1.3), except that the input consists of the two unformatted files from RECUPD rather than from STG2. The recoupled Hamiltonian matrices are diagonalized and data written to the H file, and the transformed dipole matrices written to the D files, for the total $J\pi$ symmetries requested.

Again the unformatted files from RECUPD can normally be deleted after the STGH run, as they are no longer needed, with the H and D files normally archived for later use.

11.2.5. STG4 run – Breit-Pauli

The input data is shown below.

```
-3  1  1                :IPRINT, IRAD, IPERT
1.E-3                   :AC
1.                      :RONE
1                       :IMESH
```

```
2 2.5 0.5
   2
0 0 0
0 2 1                       :JPI CASES
```

The test run is similar to the LS-coupling case (Section 11.1.4), except the input data specifies $J\pi$ rather than $SL\pi$ for the total $(N+1)$-electron symmetries. Note that IS ($=2S+1$ in LS-coupling) is specified as zero here, to indicate intermediate-coupling.

- In the case of electron scattering, the collision strengths for $e^- + Ne^+$ $({}^2P^o_{3/2} - {}^2P^o_{1/2}, {}^2P^o_{3/2} - {}^2S^e_{1/2}, {}^2P^o_{1/2} - {}^2S^e_{1/2})$, summed over the two partial waves (0^e and 1^o) for each electron energy, 2.5 and 3.0 Ry, are output in file XOMEGA:

```
    10     9     3      OMEGA
     .000000E+00    .742513E-02    .218202E+01
    2.50000E+00         3 2.679E-01 7.077E-02 4.859E-02
    3.00000E+00         3 2.768E-01 7.300E-02 5.257E-02
```

  with the partial collision strengths in file XDUMP (since IPRINT = −3):

```
  NE = 9  NZ =10    3-STATE      RA =    5.0 L2=  3 K-MATRIX
   S L PI    ENERGY     PARTIAL COLLISION STRENGTH
  0   0 0 2.500000E+0 1.617E+00 1.535E-01 1.553E+00 2.785E-02 1.379E-02 7.213E-01
  0   2 1 2.500000E+0 2.099E+00 1.144E-01 1.583E+00 4.292E-02 3.480E-02 5.480E+00
  0   0 0 3.000000E+0 1.623E+00 1.489E-01 1.595E+00 2.812E-02 1.393E-02 5.607E-01
  0   2 1 3.000000E+0 2.052E+00 1.279E-01 1.394E+00 4.488E-02 3.864E-02 7.001E+00
```

- In the case of photoionization, the first symmetry specified is taken as the initial state of Ne (i.e. 0^e), and the lowest bound state is calculated (this is the ground state of Ne, $1s^2 2s^2 2p^6$). The total photoionization cross section from this state to all possible final states (i.e. 1^o for $e^- + Ne^+$) is then calculated and output in file XSECTN:

```
    10     9     3      PHOTOIONIZATION
     .000000E+00    .742513E-02    .218202E+01
      0     0     0      1
    -.153960E+01     2
   0. PHOTON ENERGY(Ryds), CROSS SECTION (Mb) L,V:
    4.039603E+00 7.312E+00 5.605E+00
    4.539603E+00 6.749E+00 5.131E+00
```

Clearly, STG4 can be repeated using the same H and D files with different options, e.g. energy ranges and IPRINT. Note that the D files are not required if only electron scattering or bound state calculations are being performed.

11.3. Timings on a Cray Y-MP EL

The 'USER CPU Time' in seconds required to run the Breit-Pauli test case on a Cray Y-MP EL is shown below.

```
STG1      23.8635
STG2       1.8893
RECUPD     1.7808
STGH       2.1838
STG4      11.3464
```

References

[1] D.C.S. Allison, Comput. Phys. Commun. 1 (1969) 15-20.
[2] D.C.S. Allison, P.G. Burke and W.D. Robb, J. Phys. B: At. Mol. Phys. 5 (1972) 55-65.
[3] K.L. Baluja, P.G. Burke and L.A. Morgan, Comput. Phys. Commun. 27 (1982) 299-307.
[4] A.R. Barnett, D.H. Feng, J.W. Steed and L.J.B. Goldfarb, Comput. Phys. Commun. 8 (1974) 377-395.
[5] Bar-Shalom A. and M. Klapisch, Comput. Phys. Commun. 50 (1988) 375-393.
[6] K. Bartschat, Comput. Phys. Commun. 30 (1983) 383-396.
[7] K. Bartschat and P.G. Burke, Comput. Phys. Commun. 41 (1986) 75-84.
[8] K. Bartschat and N.S. Scott, Comput. Phys. Commun. 30 (1983) 369-382.
[9] K.L. Bell and N.S. Scott, Comput. Phys. Commun. 20 (1980) 447-458.
[10] K.L. Bell, P.G. Burke and A.E. Kingston, J. Phys. B: At. Mol. Phys. 10 (1977) 3117-3127.
[11] K.A. Berrington, P.G. Burke, K. Butler, M.J. Seaton, P.J. Storey, K.T. Taylor and Yu Yan, J. Phys. B: At. Mol. Phys. 20 (1987) 6379-6397.
[12] K.A. Berrington, P.G. Burke, J.J. Chang, A.T. Chivers, W.D. Robb and K.T. Taylor, Comput. Phys. Commun. 8 (1974) 149-198.
[13] K.A. Berrington, P.G. Burke, M. Le Dourneuf, W.D. Robb, K.T. Taylor and Vo Ky Lan, Comput. Phys. Commun. 14 (1978) 367-412.
[14] K.A. Berrington, W.B. Eissner, H.E. Saraph, M.J. Seaton and P.J. Storey, Comput. Phys. Commun. 44 (1987) 105-119.
[15] K.A. Berrington, A.E. Kingston and P.M.J. Sawey, J. Physique I Suppl. JPII 3 C 1 (1991) 299-302.
[16] H.A. Bethe and E.E. Salpeter, Handb. Phys. 35 (Springer, Berlin, 1957).
[17] C. Bloch, Nucl. Phys. 4 (1957) 503-528.
[18] M. Blume and R.E. Watson, Proc. R. Soc. 270 A (1962) 127-143.
[19] Burgess et al., Phil. Trans. Roy. Soc. Lond. A 266 (1970) 225.
[20] P.G. Burke, Comput. Phys. Commun. 1 (1970) 241-250.
[21] P.G. Burke and K.A. Berrington, Atomic and Molecular Processes, an R-matrix Approach (Institute of Physics Publ. Bristol) ISBN 0-7503-0199-6 (1993).
[22] P.G. Burke and W.D. Robb, Adv. At. Mol. Phys. 11 (1975) 143-214.
[23] P.G. Burke and M.J. Seaton, Methods Comput. Phys. 10 (1971) 1-80.
[24] P.G. Burke and M.J. Seaton, J. Phys. B: At. Mol. Phys. 17 (1984) L683-L687.
[25] P.G. Burke and K.T. Taylor, J. Phys. B: At. Mol. Phys. 8 (1975) 2620-2639.
[26] P.G. Burke, V.M. Burke and K.M. Dunseath, J. Phys. B: At. Mol. Phys. 27 (1994) 5341-5373.
[27] P.G. Burke, A. Hibbert and W.D. Robb, J. Phys. B: At. Mol. Phys. 4 (1971) 153-161.
[28] V.M. Burke and C.J. Noble, Comput. Phys. Commun. 85 (1995) 471-500.
[29] V.M. Burke, P.G. Burke and N.S. Scott, Comput. Phys. Commun. 69 (1992) 76-98.
[30] P.J.A. Buttle, Phys. Rev. 160 (1967) 719-729.
[31] J.J. Chang, J. Phys. B: At. Mol. Phys. 10 (1977) 3335-3339.
[32] A.T. Chivers, Comput. Phys. Commun. 6 (1973) 88.
[33] R.E.H. Clark, Comput. Phys. Commun. 16 (1978) 119-127.
[34] E. Clementi and R. Roetti, At. Data Nucl. Data Tables 14 (1974) 177-478.
[35] E.R. Cohen and B.N. Taylor, CODATA Bull. 63 (1986) .
[36] E.U. Condon and G.H. Shortley, The theory of atomic spectra (Cambridge University Press, Cambridge, England, 1935).
[37] M.A. Crees, Comput. Phys. Commun. 23 (1981) 181-198.
[38] M.A. Crees, M.J. Seaton and P.M.H. Wilson, Comput. Phys. Commun. 15 (1978) 23-83.
[39] J. Croskery, N.S. Scott, K.L. Bell and K.A. Berrington, Comput. Phys. Commun. 27 (1982) 385-401.
[40] C. Day, Comput. Phys. Commun. 13 (1977) 101-105.
[41] K.G. Dyall, I.P. Grant, C.T. Johnson, F.A. Parpia and E.P. Plummer, Comput. Phys. Commun. 55 (1989) 425-456; erratum 58 (1990) 345.
[42] W.B. Eissner, J. Physique IV 1 (1991) C3-C13.
[43] W.B. Eissner, M. Jones and H. Nussbaumer, Comput. Phys. Commun. 8 (1974) 270-306.
[44] U. Fano, Phys. Rev. 140 (1965) A67.
[45] U. Fano and G. Racah, Irreducible Tensorial Sets (Academic Press, New York, 1959).
[46] C.F. Fischer, Comput. Phys. Commun. 64 (1991) 369-419.
[47] R. Glass and A. Hibbert, Comput. Phys. Commun. 16 (1978) 19-34.
[48] C.H. Greene and M. Aymar, Phys. Rev. A 44 (1991) 1773-90.
[49] D.R. Hartree, Proc. Camb. Phil. Soc. 24 (1928) 89.
[50] A. Hibbert, Comput. Phys. Commun. 1 (1970) 359-377.
[51] A. Hibbert, Comput. Phys. Commun. 2 (1971) 180-190; erratum 6 (1973) 59-61.

[52] A. Hibbert, Comput. Phys. Commun. 9 (1975) 141-172.
[53] J.G. Hughes, F.J. Smith, K.A. Berrington, K.M. Aggarwal and M. Elder, Comput. Phys. Commun. 33 (1984) 99-103.
[54] D.G. Hummer, K.A. Berrington, W. Eissner, A.K. Pradhan, H.E. Saraph and J.A. Tully, Astron. Astrophys. 279 (1993) 298-309.
[55] M. Jones, Radiative corrections in Atomic Structure Calculations, Ph.D. Thesis, London (1971).
[56] M. Jones, Phil. Trans. Roy. Soc. (London) 277 (1975) 587-622.
[57] A.M. Lane, J. Phys. B: At. Mol. Phys. 19 (1986) 253-257.
[58] J.C. Light and R.B. Walker, J. Chem. Phys. 65 (1976) 4272-4282.
[59] D.L. Moores, Comput. Phys. Commun. 2 (1971) 360-367.
[60] C.J. Noble and R.K. Nesbet, Comput. Phys. Commun. 33 (1984) 399-411.
[61] D.W. Norcross, Comput. Phys. Commun. 1 (1969) 88-96.
[62] P.H. Norrington and I.P. Grant, J. Phys. B: At. Mol. Phys. 14 (1981) L261-L267.
[63] K. Onda, D.G. Truhlar and M.A. Brandt, Comput. Phys. Commun. 21 (1980) 97-108.
[64] Opacity Project Team, The Opacity Project Volume 1 (Institute of Physics Publ. Bristol) ISBN 0-7503-0288-7 (1994).
[65] G. Racah, Phys. Rev. 61 (1942) 537.
[66] W.D. Robb, Comput. Phys. Commun. 1 (1970) 457-464.
[67] W.D. Robb, Comput. Phys. Commun. 6 (1973) 132-148.
[68] S.A. Salvini, Comput. Phys. Commun. 27 (1982) 25-37.
[69] H.E. Saraph, Comput. Phys. Commun. 3 (1972) 256-268.
[70] H.E. Saraph, Comput. Phys. Commun. 15 (1978) 247-258.
[71] N.S. Scott and P.G. Burke, J. Phys. B: At. Mol. Phys. 13 (1980) 4299-4314.
[72] N.S. Scott and A. Hibbert, Comput. Phys. Commun. 28 (1982) 189-200.
[73] N.S. Scott and K.T. Taylor, Comput. Phys. Commun. 25 (1982) 347-387.
[74] M.J. Seaton, Rep. Prog. Phys. 46 (1983) 167-257.
[75] M.J. Seaton, J. Phys. B: At. Mol. Phys. 18 (1985) 2111-2131.
[76] M.J. Seaton, J. Phys. B: At. Mol. Phys. 19 (1986) 2601-2610.
[77] M.J. Seaton, J. Phys. B: At. Mol. Phys. 20 (1987) L69-L72.
[78] M.J. Seaton, J. Phys. B: At. Mol. Phys. 20 (1987) 6363-6378.
[79] L.F. Shampine and M.K. Gordon, Computer Solution of Ordinary Differential Equations, (Freeman, New York, 1975).
[80] P. Shorer, J. Phys. B: At. Mol. Phys. 13 (1988) 1921-1929.
[81] J.C. Slater, Quantum Theory of Atomic Structure, Vol. 1 (McGraw-Hill, New York, 1960).
[82] J. Tennyson and C.J. Noble, Comput. Phys. Commun. 33 (1984) 421-424.
[83] Wigner and Eisenbud Phys. Rev. 72 (1947) 29-41.
[84] J.H. Wilkinson, Comput. J. 3 (1960) 23-27.
[85] Yu Yan and M.J. Seaton, J. Phys. B: At. Mol. Phys. 18 (1985) 2577-2585.