

UKRmol: a low-energy electron- and positron-molecule scattering suite

J.M. Carr¹, P.G. Galiatsatos², J.D. Gorfinkiel^{3,a}, A.G. Harvey⁵, M.A. Lysaght³, D. Madden², Z. Mašín³, M. Plummer⁴, J. Tennyson², and H.N. Varambha²

¹ Cambridge University Centre for Computational Chemistry, University Chemical Laboratory, Lensfield Road, CB2 1EW Cambridge, UK

² Department of Physics and Astronomy, University College London, WC1E 6BT Gower St., London, UK

³ Department of Physical Sciences, The Open University, Walton Hall, MK7 6AA Milton Keynes, UK

⁴ Computational Science and Engineering Department, STFC Daresbury Laboratory, Daresbury, WA4 4AD Warrington, UK

⁵ Max-Born-Institute, Division B, Max-Born-Str. 2A, 12489 Berlin, Germany

Received 9 November 2011 / Received in final form 14 December 2011

Published online 1 March 2012 – © EDP Sciences, Società Italiana di Fisica, Springer-Verlag 2012

Abstract. We describe the UK computational implementation of the *R*-matrix method for the treatment of electron and positron scattering from molecules. Recent developments in the UKRmol suite are detailed together with the collision processes it is enabling us to treat.

1 Introduction

The properties of cool plasmas are dominated by electron – molecule collision processes. For example, in our own atmosphere the process of dissociative electron attachment (see Sect. 5) was originally proposed to explain the ionisation balance in the ionosphere [1]. Similarly aurora [2] and lightning [3] are natural atmospheric phenomena which depend on electron – molecule collisions to drive processes such as electronic excitation.

Most of the Universe is a weakly ionised plasma and electron – molecule collision processes are very important in the cooler parts of it. As well as planetary auroras [4] mentioned above, electron collisions initiate much of the chemistry found in the interstellar medium [5] and shocks [6]. Planetary nebulae [7], which are hotter interstellar regions, are also influenced by processes involving electron molecule collisions.

In a seminal experimental study Boudaïffa et al. [8] showed that most of the energy deposited in cells by ionising radiation is channelled into the production of free secondary electrons with energies below 20 eV and that reactions of such electrons induce substantial yields of single- and double-strand breaks in DNA. These strand breaks were associated with electrons trapped in quasi-bound or resonance states on the basic components of the DNA. These observations have sparked a host of studies, both theoretical and experimental, on electron collisions with DNA (and RNA) components [9].

The use of controlled plasmas for processing has been a major driver of modern technology. This and other

technologies rely on electron – molecule collisions. Everyday examples include the spark plugs that ignite internal combustion engines, electric lighting and many laser processes. Electron collisions initiate interesting chemistries which are exploited in processes based on plasma etching. The importance of electron – molecule collisions, particularly at low energy, has of course stimulated significant work both experimentally [10] and theoretically [11].

Low energy electron – molecule collisions are characterised by a number of subtle interactions and many possible outcomes. It is perhaps not surprising that the prototypical processes (listed in Sect. 5) are mainly sensitive to collisions in this energy range. The ionisation threshold marks the transition from the low to the intermediate energy regime. Standard close-coupling approaches do not give reliable results at intermediate energies.

There are a number of ab initio theoretical methods available for treating low-energy electron – molecule collisions, most of which have also been applied to positron scattering. Competitive with the *R*-matrix-based methods, one of which is the subject of the present article, are those based on the Kohn variational method [12,13], and the Schwinger variational principle [14–16]. The whole *R*-matrix method is the subject of a recent monograph by Burke [17] and its specific use for electron – molecule collisions has recently been reviewed by one of us [18].

The UK-molecular *R*-matrix codes, which are the subject of the present article, are available as free-ware. Morgan et al. [19] published an account of the codes shortly after the original implementation of a Gaussian type orbital (GTO) electron polyatomic molecule code [20]. Since then the codes have undergone a comprehensive re-write and modernisation, have a number

^a e-mail: J.Gorfinkiel@open.ac.uk

of important algorithmic changes and have significantly increased functionality. The suite is now called UKRmol. It is therefore timely to report on the latest versions of the code and the processes we are now able to investigate. This is what we do here.

2 The R-matrix method

The R -matrix method and its application to electron – molecule collisions have been described in detail elsewhere [17,18]. Therefore, only a brief summary is presented in this section, with the emphasis on those parts of the calculation whose computational implementation has been recently updated. We work initially within the fixed-nuclei approximation, in which the nuclei of the molecule are held fixed during the collision, and solve the electronic part of the problem. The information thus obtained can then be used in calculations that incorporate nuclear motion. For example, rotational excitation can be described within the adiabatic-nuclei rotation approximation [21].

The basic idea of the R -matrix method is the partition of configuration space into an inner and an outer region, separated (in most practical cases) by a sphere of radius a . The electronic density of the target states of interest must be contained inside the inner region. Exchange and correlation between all electrons must be taken into account in this region (only correlation effects with a positronic projectile, although these are stronger). The eigenfunctions of the total Hamiltonian are expanded in the following way:

$$\begin{aligned} \Psi_k^{N+1}(x_1 \dots x_{N+1}) = & \mathcal{A} \sum_{ij} a_{ijk} \phi_i(x_1 \dots x_N) u_{ij}(x_{N+1}) \\ & + \sum_i b_{ik} \chi_i(x_1 \dots x_{N+1}) \end{aligned} \quad (1)$$

where \mathcal{A} is the antisymmetrization operator, $u_{ij}(x_{N+1})$ are ‘continuum’ orbitals describing the scattering electron and x_n are the spatial and spin coordinates of electron n . The continuum orbitals are generated by orthogonalizing to the target molecular orbitals (both those used to describe the target states and extra orbitals used for short-range correlation and polarization; see below) a set of GTOs centred on the centre of mass of the system. The use of GTOs to describe the continuum simplifies the integral calculation and ensures these are determined to high precision. However, it also limits the size of the R -matrix sphere (to around 15–18 a_0 [22]) and therefore makes it difficult to include diffuse (Rydberg) states in the calculation. ϕ_i are wavefunctions describing true or pseudostates of the target molecule. χ_i are multi-centre quadratically integrable functions known as L^2 functions, constructed from the target occupied and virtual molecular orbitals (MOs); they are used to represent short range correlation and polarization effects. The coefficients a_{ijk} and b_{ik} are obtained by diagonalizing the matrix of order M :

$$(\Psi_k^{N+1} | \hat{H}^{N+1} - \hat{L} | \Psi_{k'}^{N+1}) = E_k \delta_{kk'} \quad (2)$$

with \hat{H}^{N+1} the $(N+1)$ -electron Hamiltonian and \hat{L} the Bloch operator [17].

Using the inner region wavefunctions and their associated eigenvalues, together with information on the target molecule to define channels, the R -matrix is built at the boundary between the two regions:

$$R_{ij}(E) = \frac{1}{2a} \sum_k \frac{w_{ik}(a) w_{jk}(a)}{E_k - E} \quad (3)$$

where E_k , known as R -matrix poles, are the eigenvalues from equation (2) and w_{ik} are:

$$w_{ik} = \langle \phi_i Y_{l_i m_i} \Xi_{\frac{1}{2}} | \Psi_k^{N+1} \rangle. \quad (4)$$

The product $\phi_i Y_{l_i m_i} \Xi_{\frac{1}{2}}$ defines the channels. The function $\Xi_{\frac{1}{2}}$ is a one half electron spin function and $Y_{l_i m_i}$ are spherical harmonics. The integration to obtain the w_{ik} is performed over all space-spin coordinates except the radial coordinate of the scattering electron.

When the matrix to diagonalize is very large (when the number of configurations goes into the hundreds of thousands) an alternative procedure becomes necessary. The partitioned R -matrix theory [23] requires only the lowest m eigenpairs of equation (2) to be explicitly used when building the R -matrix; the remaining terms are implicitly allowed for using an error correction procedure. The expression for the R -matrix becomes:

$$\begin{aligned} R_{ij}(E) = & \frac{1}{2a} \sum_{k=1}^m w_{ik}(a) w_{jk}(a) \left(\frac{1}{E_k - E} - \frac{1}{E_0 - E} \right) \\ & + \delta_{ij} \left(\frac{s_i}{E_0 - E} + R_{ij}^C \right). \end{aligned} \quad (5)$$

Details of the parameters E_0 , s_i and the error correction term R_{ij}^C can be found in [24].

In the outer region, exchange and correlation can be neglected and a single-centre multipole expansion of the projectile-molecule interaction potential suffices. This leads to a set of coupled differential equations for the functions describing the radial behaviour of the projectile.

The set of differential equations is solved by propagating the R -matrix from the boundary to some large radial distance (chosen such that the non-Coulombic potential can be neglected beyond it). The radial wavefunctions are then matched to asymptotic solutions of known form. In this way, K -matrices are obtained.

The set of values chosen for l and m of the spherical harmonics describing the angular part of the ‘continuum’ GTOs determines the partial wave expansion. This expansion converges rapidly for most cases; inclusion of just a few partial waves (normally $l \leq 4$) is usually sufficient. However, this is not the case when the target has a permanent dipole moment, in which case we need to use the dipole Born approximation to account for higher partial waves [25].

The transition moments for the multipole expansion are calculated from the wavefunctions of the target states of interest in the first step of our calculations. For this purpose, one must define a model for the description of the target. That is, one must choose a basis set for the

molecular orbitals, a method for generating said orbitals and a configuration interaction (CI) model to produce the multielectronic wave functions. The basis sets used are standard (but sometimes adapted) quantum chemistry basis sets. To generate the orbitals, we run a Hartree-Fock self consistent field calculation (HF-SCF), produce (state-averaged if needed) pseudo-natural orbitals or feed the program a set of orbitals obtained from an external quantum chemistry code (for example, by means of a CASSCF calculation).

Simple static-exchange or static-exchange plus polarization scattering models employ a single configuration wave function to describe the ground state of the target. Otherwise the CI model is chosen so that several electronic states of the target are well described using a single orbital set. The quality of the target description is checked against available experimental and computational data (excitation energies, dipole and quadrupole moments, etc.).

For small target molecules, a CAS (complete active space: all the possible configurations arising from distributing the electrons among the orbitals in the active space are built) model ensures good balance between the target and target + projectile calculations [26]. For larger molecules, the need to include at least single excitations from the active space into a space of virtual orbitals makes it difficult to ensure balance a priori.

In many cases, it is appropriate to augment the close-coupling expansion of equation (1) with a set of wave functions ϕ_i that represent states that are not true eigenstates of the target. This is known as the *R*-matrix with pseudostates method (RMPS), developed initially in the context of electron – atom collisions to treat the intermediate energy range [27]. The method was adapted to treat near-threshold ionization in molecules [28,29] but it has proven fundamental for the accurate representation of polarization effects: for example, in positron collisions with non-polar targets, inclusion of pseudostates is essential for the description of the process (see Sect. 4.4). The pseudostates are obtained by diagonalizing the target electronic Hamiltonian in a suitable basis of configurations. A new set of orbitals, called pseudo-continuum orbitals (PCOs), is introduced in the calculations. These PCOs are expanded in terms of an even-tempered basis set [30] of GTOs centred at the centre of mass of the system. Then, on top of the usual configurations employed in the target description (where all the electrons occupy MOs) another set of configurations is included in which one of the target electrons occupies a PCO.

3 The suite

The UKRmol suite is divided into two subsets of programs: UKRmol-in¹ and UKRmol-out². The first set encompasses the programs required to run the inner region

¹ <http://ccpforge.cse.rl.ac.uk/gf/project/ukrmol-in/>

² <http://ccpforge.cse.rl.ac.uk/gf/project/ukrmol-out/>

part of the calculation; most of these, together with some additional programs, are run to obtain the target properties (and the orthogonalized set of target orbitals). The second encompasses the programs required to run the outer region part of the calculation. An implementation of the inner region set of programs based on the use of Slater type orbitals for the description of the target and numerical continuum functions, is available to treat collisions with diatomic molecules [19,31]. This suite replaces all the programs related to the calculation of integrals and orbitals but uses the same programs for the construction and diagonalization of the Hamiltonian (see below). We will not describe the details of this suite here since it is largely unchanged since the previous report [19].

The polyatomic UKRmol inner region codes are based on the Sweden-Molecule quantum chemistry code of Almlöf and Taylor [32]. Figure 1 shows a flowchart describing the calculation of the target properties. The programs involved perform the following tasks:

SWMOL3 calculates the integrals between GTO basis functions that will be used later on. This program takes as input (in addition to a basis set) the geometry of the molecule and the point group it belongs to.

SWORD orders the calculated integrals.

SWFJK and SWSCF are run to obtain Hartree-Fock Self-Consistent-Field molecular orbitals. Nowadays, particularly for calculations involving several electronic states, it is usual to use orbitals obtained from more sophisticated methods (e.g. CASSCF) and external standard quantum chemistry codes. The program MPOUTRD has been developed to interface between MOLPRO [33] and the UKRmol suite.

SWEDMOS performs the orthogonalization of the orbitals to be used in the calculation (which may not be the whole set generated, for example, in the HF-SCF step). The Schmidt technique is used for the orthogonalization.

SWTRMO performs the transformation of the integrals labelled according to atomic orbitals to a set labelled over molecular orbitals.

CONGEN generates the configuration state functions (CSFs) to be used in the CI description of the target (even if a single-determinant description is chosen, this and the following program need to be run). The user input required in this program are the parameters defining the specific configurations to be built.

SCATCI builds the Hamiltonian matrix for the system using CSFs from CONGEN and the integrals transformed by SWTRMO. This Hamiltonian is then diagonalized to obtain the number of required eigenpairs.

GAUSPROP calculates property integrals for GTOs required by the next program.

DENPROP builds the density matrices for the target state(s) included in the calculation and determines the transition moments. This program can also calculate the polarizability of the target; although not used explicitly in the calculations, it can give an idea of the quality of the polarization description in close-coupling calculations [34].

Figure 2 shows the flowchart of an inner region calculation. The same (but not all) programs as in the target

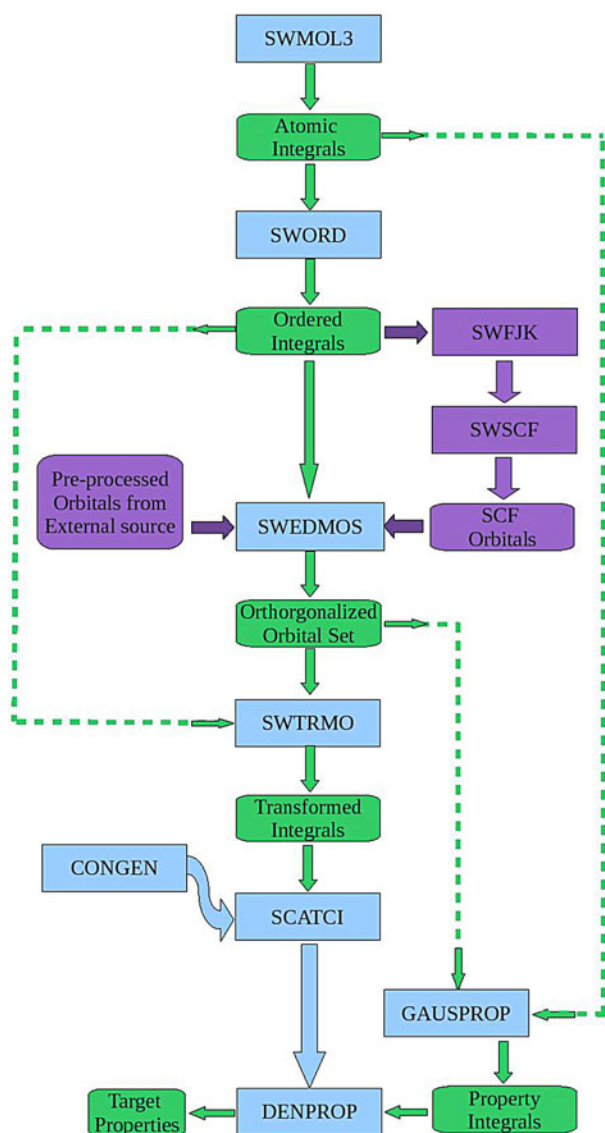


Fig. 1. (Color online) Flowchart for the target part of the calculation. The blue boxes indicate programs in the suite. The green boxes indicate the main output/input of the different programs. The purple boxes indicate alternative options to produce molecular orbitals in the calculation. Full arrows indicate the flow of the calculation, in particular those programs/input that feed into the following program in the suite. The dashed lines coming out of the green boxes indicate input for programs.

calculations are used with the addition of GAUSTAIL. This program calculates the ‘tail’ integrals, integrals between the boundary of the R -matrix sphere and ∞ [20], that are then subtracted in SWORD from those generated in SWMOL3. GAUSTAIL requires as input the radius of the R -matrix sphere. The choice of radius also defines the continuum basis set that needs to be input to SWMOL3 in the inner region run.

When SWEDMOS is run in an inner region calculation, it performs two orthogonalizations: first, of the continuum basis set to the target molecular orbitals included

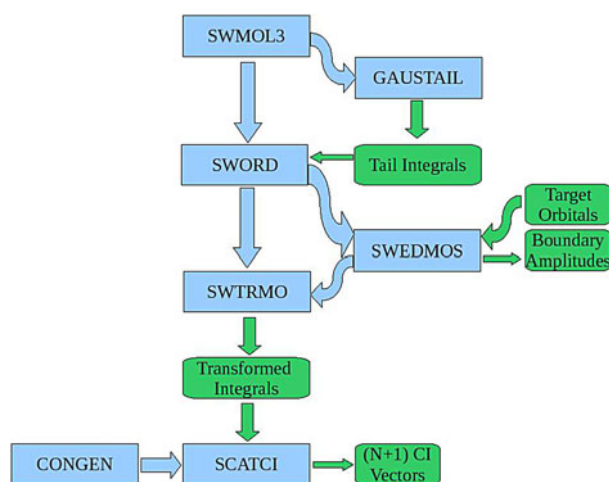


Fig. 2. (Color online) Flowchart for the inner part of the calculation. Blue boxes indicate programs in the suite. Green boxes indicate the main output/input of the different programs. The arrows indicate the flow of the calculation, in particular those programs/input that feed into the following program in the suite.

in the calculation (and generated and/or orthogonalized in the target run). Then, the continuum orbitals are symmetric-orthogonalized among themselves. The program also generates the raw boundary amplitudes, that is the amplitudes of the continuum orbitals at the boundary between the regions (the amplitudes of the target orbitals are negligible by construction) necessary to build the amplitudes defined in equation (4).

In an inner region run, all eigenpairs are in principle required in SCATCI and the CSFs built in CONGEN are of the type indicated in equation (1). Since the Hamiltonian for the system is block-diagonal (with P blocks, corresponding to all possible space-spin symmetries of the $N+1$ system), P runs of CONGEN and SCATCI are performed. Details of the diagonalization are presented in Section 4.3.

Figure 3 shows the flowchart for the outer region calculation. As is the case for CONGEN and SCATCI, the outer region suite is run for each space-spin symmetry of the system. The programs involved are:

SWINTERF is an interface program that takes input from the target run (target properties) and inner region run (raw boundary amplitudes and eigenpairs) and produces two files containing channel data and data needed to build the R -matrix at the boundary between the regions. The user should indicate in the input how many of the target states included in the close-coupling expansion of the inner region are to be retained in the outer region calculation.

RSOLVE generates the K -matrices by building the R -matrix at the boundary between the regions and propagating it to a radial distance (normally a few tens of bohr) [35] where an asymptotic expansion provided by module CFASYM [36] can be used and the results then matched to asymptotic Coulomb or Bessel functions [37]. The scattering energies for which the scattering information is required must be specified here. Once the

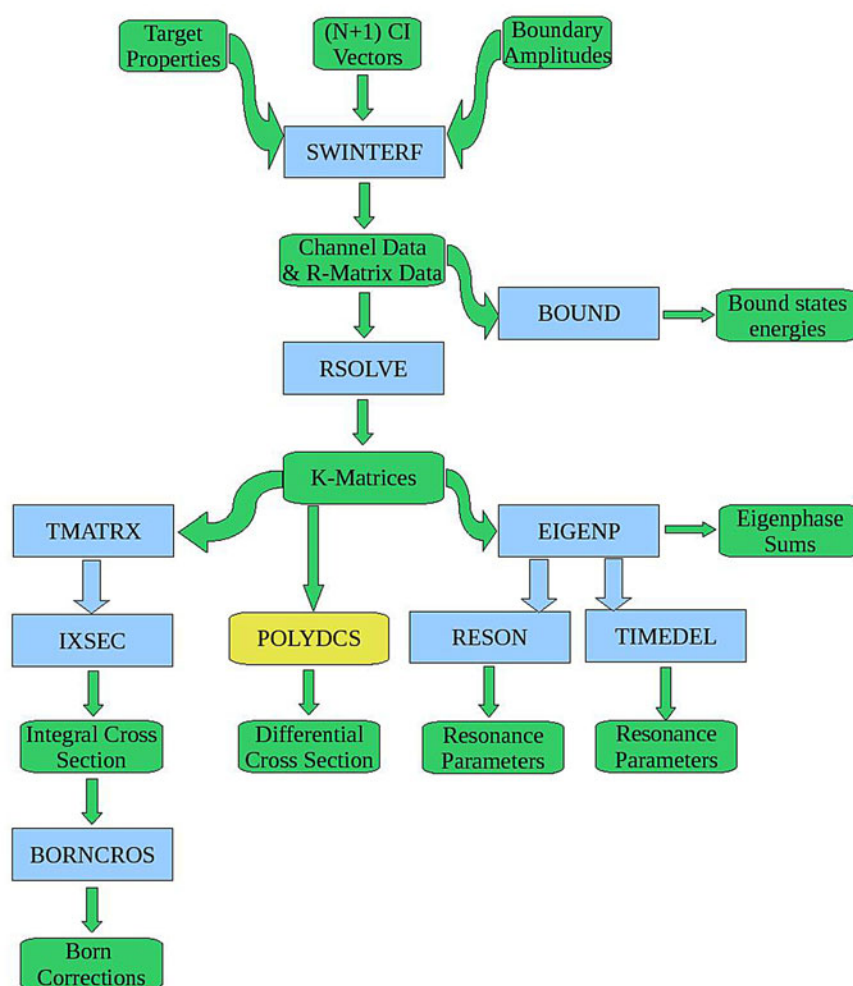


Fig. 3. (Color online) Flowchart for the outer part of the calculation. Blue boxes indicate programs in the suite. Green boxes indicate the main output/input of the different programs. The arrows indicate the flow of the calculation, in particular those programs/input that feed into the following program in the suite. PolyDCS is not part of the UKRmol suite; it was developed by Sanna and Gianturco [40].

K -matrices are available several programs can be run to obtain different scattering quantities and observables.

EIGENP calculates the eigenphase sum by diagonalizing the K -matrix.

RESON [38] uses the eigenphase sums to find and fit resonances to a Breit-Wigner profile in order to determine their energy and width. TIMEDEL [39] does the same, but using the time-delay method. Both programs work recursively in that they analyse the eigenphase sums for possible resonances and call RSOLVE at a suitable grid of energies before performing a fit.

TMATRX determines T -matrices from the K -matrices via simple matrix operations.

IXSEC calculates integral cross sections from T -matrices.

The K -matrices can be input into POLYDCS [40] to obtain electronically elastic differential cross sections, rotationally resolved integral cross sections and, also, to complete the partial wave expansion through the inclusion

of a Born-type correction in the case of polar target molecules.

BORNCROS adds up the contribution of the different symmetries to obtain the integral cross section and also calculates an approximate Born correction both for electronically elastic and inelastic cross sections [41].

BOUND [42] finds bound states of the $N + 1$ system.

There are a number of programs, not included in the flowcharts, that are only run sporadically. For example, PSN is a program designed to obtain pseudonatural orbitals from the diagonalization of density matrices obtained in DENPROP. The program can state average these orbitals using a set of user provided weights. Similarly CDENPROP [43] is a new code that calculates transition moments for the inner region wavefunctions.

For the inner region calculation, it is necessary to have a GTO ‘continuum basis set’. In order to obtain one we use the following modules:

NUMCBAS evaluates Bessel and Coulomb functions within the R -matrix sphere. GTOBAS [44] fits GTOs to

the continuum functions generated by NUMCBAS. The exponents of GTOs are optimized using the method of Nestmann and Peyerimhoff [45].

In the outer region, the following programs are also available:

ALIGN [46] enables the treatment of collisions with oriented molecules.

MCQD [47] determines the multichannel quantum defect for $N + 1$ states.

ROTIONS [48] calculates rotational excitation cross sections for linear ions.

ZEFF calculates the energy-dependent annihilation parameter Z_{eff} in the case of positron collisions using integrals computed in the inner region by GAUSDELTA [49].

4 Current developments

4.1 General strategy and community-based good practice

The UKRmol codes are registered as projects on CCPForge³, a collaborative software development environment tool with overall administration and assistance by the Software Engineering Group (SEG) of the UK Science and Technology Facilities Council. CCPForge provides an on-line repository for the codes, allowing version control via Subversion⁴, user and developer forums and mailing lists, bug reporting, tracking tools and more. The UKRmol molecular R -matrix suite can be downloaded from CCPForge as official user releases and as developers' packages, with associated test suites of example calculations. The development of official test suites has enabled us to test ported code, verify new code and validate backwards-compatibility.

As part of the re-engineering of the UKRmol suite, it was decided that new coding would be subject to agreed 'best practice' guidelines and rules to enable it to be long-lasting, portable and understandable by large and disparate groups of often transient developers [50]. It was also decided that the existing 'legacy' molecular R -matrix codes [19], some of which date back decades, should be upgraded, as far as possible, to be consistent with these new guidelines. The outer region molecular R -matrix codes were comprehensively rewritten in a modular form by Morgan in the 1990s [19], so the main developments of UKRmol relate to the inner region codes.

The suite has been re-engineered using a hierarchy of high-level scientific 'functional' modules and lower-level 'utility' and 'external interface' modules used by the functional modules as described in Section 4.2. All modules are subject to agreed interfacing guidelines to avoid future problems with backwards compatibility. The rules and guidelines were influenced by the practical experience of colleagues who are developers of the materials science code CASTEP [51], which has been set up using similar

'best practice' ideas for over a decade and which is currently the most downloaded code on CCPForge.

In addition the massively-parallel electron-atom outer region code PFARM [52] has been adapted to accept input data from UKRmol-in via a new UKRmol-out module, QBinterface. It is currently undergoing testing as an alternative option in UKRmol-out: a full description of this development will appear in due course.

4.2 Coding upgrades

The aim of the UKRmol-in software re-engineering project was to produce code compliant with the Fortran 95 standard [53] and that has a clear, unified, modular structure. As such, the updated software is easy to maintain, port to different computational platforms, and extend by adding new functionality in the future.

The main translation and reformatting of the source code employed the SPAG tool from the plusFORT suite of utilities for Fortran programmers⁵. SPAG was configured to perform the following:

- Conversion from implicit to explicit typing: all variables must be explicitly declared, and subprograms contain the IMPLICIT NONE statement.
- Translation of DOUBLE PRECISION to REAL (kind-parameter-name) in declarations.
- Rewriting of declarations in Fortran 95 style, for instance adding INTENT statements.
- Translation of DO loops to Fortran 95 style (use of END DO, EXIT and CYCLE).
- Translation of END statements to Fortran 95 style END SUBROUTINE, END FUNCTION or END PROGRAM, with the subprogram name appended.
- Old-fashioned control constructs, such as GOTO and arithmetic IF, are replaced without changing the logic of the code.
- Application of a consistent spacing policy within certain statements (e.g. END DO) and around various operators.
- Application of consistent indentation patterns.
- Removal of clutter, such as unused variables.

The changes are conservative; some are cosmetic, in order to improve the readability of the code, but others make use of important security features available in Fortran 95 (such as explicit typing and the INTENT attribute for subprogram dummy arguments). Further changes were made in a non-automated manner. Statement functions and ENTRY statements, which are obsolete, were replaced using appropriate modern concepts. Variables stored in COMMON blocks were moved into modules, from which the variables are available via USE association.

In addition to changes made within each program in the suite in turn, the data structures and organisation of the source code between programs were also treated. A policy of modularisation/abstraction was introduced, for

³ <http://ccpforge.cse.rl.ac.uk/gf/>

⁴ <http://subversion.apache.org/>

⁵ <http://www.polyhedron.com/pf-plusfort0html>

both global data and global procedures, in order to ensure consistency between different programs and localise the changes needed when updating the data or the procedures, thus improving portability and maintainability. Routines duplicated in several programs were consolidated into a single ‘utility’ module procedure. Modules containing data needed by many of the programs, such as parameters corresponding to physical constants, were created with the variables they contain available via USE association. A global treatment of the precision and/or range of numeric data was implemented in such a way, by defining a global parameter used in specifying the kinds of constants, in type conversion functions, and as the kind selector in the corresponding type declarations. Also, high-level, scientific ‘functional’ routines no longer call low-level subprograms directly, but only via interfacing (wrapping) routines that arrange the required data in the necessary forms and call the appropriate subprogram. Low-level subprograms that require interfacing include routines from external maths libraries, Message Passing Interface routines and variables for parallel communication, timers, I/O functions and integer-packing routines specific to the UKRmol programs. The interfacing subprograms are collected into modules, at least one for each type of functionality (e.g. maths, communications, etc.).

All changes to the code were checked thoroughly before being committed to the version control repository: unit tests were created for specific functionalities, and the UKRmol-in testing suite of calculations was employed for regression testing.

4.3 Hamiltonian construction and diagonalization

The construction and diagonalisation of the inner region Hamiltonian matrix is the major computational bottleneck in the calculations. These steps rely on a Hamiltonian construction algorithm tailored to be efficient for scattering wavefunctions of the form of equation (1) [54] and also to deal with phase issues which arise from combining N and $N + 1$ electron wavefunctions [55]. The construction and diagonalization of both the N - and $(N+1)$ -electron Hamiltonian in the R -matrix inner region occurs within the SCATCI program. It should be noted that SCATCI deals with compound configuration state functions (CSFs) which do not actually obey Slater’s rules.

The computational demands of diagonalising the Hamiltonian matrices encountered in the code mean that several diagonalization methods are implemented [18]:

- For small Hamiltonian matrices, when only a few eigenpairs are required (that is, for target calculations) a Davidson diagonalization procedure is used [56].
- When all eigenpairs are required and the matrix is not too big, a full diagonaliser technique can be used; LAPACK [57] divide and conquer full matrix diagonalisation routine dsyevd is currently employed.
- For large to very large Hamiltonians, particularly in inner region calculations, the partitioned R -matrix technique is used [24] and iterative diagonalizers [58] can be

employed to calculate a small percentage of the eigenpairs.

Upgrade work on SCATCI has focused on parallelising subroutines involved in both the construction and diagonalization of the $(N+1)$ -electron Hamiltonian. For problems in which the order of the Hamiltonian Matrix (defined in equation (2)) is large ($M > 100\,000$), over 90% of UKRmol-in compute time is spent within subroutines associated with its diagonalization. In choosing numerical methods to diagonalize the matrix, we note the fact that the partitioned R -matrix method requires less than 5% of eigenpairs for large matrices. In the majority of cases, this matrix is extremely sparse (>99% sparsity, see [59] for example). For these reasons we choose to employ iterative Krylov-subspace-based eigensolvers to compute eigenpairs, due to their efficiency over direct methods for such problem types. Since the kernel method of such eigensolvers is sparse Matrix-Vector (spMV) multiplication, they also lend themselves to parallelization on both shared memory and distributed memory architectures.

Our initial efforts have focused solely on parallelizing subroutines associated with the diagonalization. The current serial version of SCATCI interfaces with the well known ARPACK library [58] which is based upon an algorithmic variant of the Arnoldi/Lanczos process called the Implicitly Restarted Arnoldi/Lanczos Method (IRAM) [60]. The ARPACK library adopts a reverse communication interface which, in practice, means a spMV multiplication mechanism needs to be provided when interfacing to the library. In the current serial version of SCATCI, spMV multiplication is implemented within the subroutine MKARP. Within MKARP, the lower triangular part of the symmetric matrix of equation (2) is read into RAM from a sequential access file (where it is stored in an unordered coordinate (COO) symmetric storage format). In order to obtain the eigenvalues and eigenvectors, a loop until convergence is performed with successive calls to ARPACK’s DSAUPD subroutine, which requires the reverse communication interface and where the spMV multiplication mechanism is provided by us.

Most of the compute time within IRAM is spent in the spMV multiplication stage and therefore our initial efforts have focused on parallelizing this stage of the algorithm. To this end, we have introduced simple modifications to the existing serial spMV multiplication mechanism using OpenMP DO directives. Since many of the subroutines within ARPACK interface with LAPACK and BLAS subroutines, multi-threaded parallelism was also introduced to these stages within the IRAM algorithm by linking to optimized threaded LAPACK and BLAS libraries such as Intel’s Math Kernel Library (MKL) with the inclusion of relevant flags at compile time.

As the OpenMP parallelized version of SCATCI is limited to a single shared memory compute node, we have also recently initiated the development of a scalable version of SCATCI that can run on more than a single compute node on distributed memory architectures. To this end we are currently interfacing SCATCI with the Scalable Library for Eigenvalue Computations (SLEPc) [61]. SLEPc is a

software library for the solution of large scale sparse eigenvalue problems on parallel computers. The SLEPc library is based on PETSc (Portable, Extensible Toolkit for Scientific Computation) [62] data structures and it employs the MPI standard for message-passing communication. The three basic abstract PETSc data objects are index sets, vectors and matrices. Built on top of this foundation are various classes of solver objects, which encapsulate virtually all information regarding the solution procedure for a particular class of problems, including various options such as convergence tolerances, etc. The Eigenvalue Problem Solver (EPS) is the main object provided by SLEPc. It is used to specify an eigenvalue problem, either in standard or generalized form, and provides uniform and efficient access to all of the eigensolvers included in the package. There are several methods available for solving eigenvalue problems within SLEPc including the Arnoldi/Lanczos method, the Jacobi-Davidson method and the Krylov-Schur method, the latter being the default method (and the one we currently use) due to its very effective restarting technique.

Rather than writing the Hamiltonian to file, as is the case in the serial version of SCATCI, in this MPI-based version of SCATCI the Hamiltonian is built in parallel on separate compute cores with each core subsequently inserting its local sparse Hamiltonian matrix elements directly into a PETSc matrix object which is subsequently passed to EPS as the operator that defines the eigenvalue problem. Various options are then set for a customized solution, the problem is then solved and finally the solution is retrieved. All interfacing with PETSc and SLEPc within SCATCI is done via a set of Fortran calling sequences within a newly introduced module known as *SlepMod*.

Testing of the new MPI version of SCATCI has recently been carried out on the UK's HECToR Cray XE6 supercomputer⁶ using phosphate as a target molecule ($M \sim 120\,000$). For this small test case the new MPI-based version of SCATCI has been shown to scale to 64 compute cores. Current development work is focusing on balancing computational load over the compute cores, a non-trivial task due to both the unstructured and unpredictable nature of the matrix.

4.4 Positron scattering

The diatomic version of the code was adapted for positron – molecule collisions some time ago [63–65] although the results were generally disappointing. More recently positron scattering has been re-implemented as part of the polyatomic code [66]. In principle both implementations are computationally straightforward; the changes require the deletion of exchange integrals, a sign change for electron – positron ‘one-electron’ integrals and a relaxation of the rules to allow the positron to occupy orbitals already filled with electrons. In addition care must be taken with

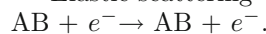
spin coupling to ensure that the positron does not couple electronic states of different spins.

In practice positron collisions are harder to treat than electron ones as the attractive positron-electron correlation effects are not well represented using the sort of standard close-coupling procedure which has provided good results for electron collisions [63]. The real advance in the positron collision calculations has been the use of the RMPS method which gives a greatly improved treatment of positron-electron correlation effects. It has been found that provided high- l orbitals are used as part of the pseudostate basis set, then very significantly improved results can be obtained. So far the RMPS method and the new code have only been used to consider positron collisions with H_2 [49], in a calculation which also obtained good results for positron annihilation, and with acetylene [59]. The use of extended RMPS basis sets make these positron calculations particularly expensive computationally and further advances will rely on the use of improved diagonalisation procedures as discussed above.

5 Processes we study: some examples of recent work

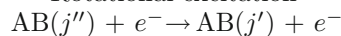
The processes that need to be considered in electron molecule scattering are listed below in approximate order of increasing electron impact energy. We have also provided examples of recent calculations performed with the UKRmol suite. The processes have been studied for both diatomic and polyatomic molecules unless otherwise stated. AB stands for a generic polyatomic molecule.

Elastic scattering



Integral and differential elastic cross sections are calculated routinely for small molecules [67], molecules of biological interest [68] and small molecular clusters [69].

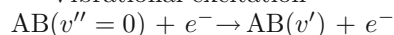
Rotational excitation



where j denotes the rotational state of the target.

Rotational excitation cross sections have been determined for a number of small linear and non-linear molecules [67,70,71] within the adiabatic nuclear rotation approximation. Spherical, symmetric and asymmetric rotors have been treated.

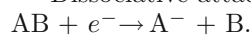
Vibrational excitation



where v denotes the vibrational state of the target.

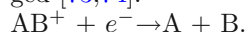
Cross sections for vibrational excitation have so far been limited to diatomic molecules (e.g. [72]).

Dissociative attachment: AB is neutral



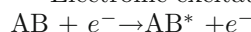
⁶ <http://www.hector.ac.uk>

Dissociative recombination: AB is positively charged [73,74].



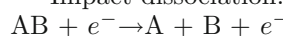
Calculations to determine the characteristics (position and width) of resonances, necessary for the study of the dissociative electron attachment and recombination processes are carried out regularly both for small and larger molecules [75–77].

Electronic excitation



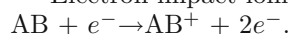
where the asterisk is used to denote an electronically excited state. Integral cross sections are a routine outcome of coupled states calculations e.g. [78–81].

Impact dissociation:



which in practise usually goes via electronic excitation has been studied extensively for the hydrogen molecule [82] and its isotopologues [83], as well as for polyatomics in reduced dimension [84].

Electron impact ionisation



Cross sections for near-threshold electron ionization have been calculated using the RMPS method for two-electron molecules [28,29]. The related process of electron impact electron detachment from anions has also been studied [85,86].

For positron collisions, work is currently limited to below the positronium formation threshold [49,59] and deals with elastic scattering (providing both integral and differential cross sections) and annihilation. Differential cross sections are important not so much for direct comparisons with experiment but to aid the measurement of elastic cross sections due to experimental issues with collection angles [87].

6 Conclusions

The developments to the UKRmol suite presented here are enabling us to explore new processes and study bigger, more complex targets. However, there is still work to be done. One of the limitations of the current implementation is that only calculations for relatively small ($\leq 18a_0$) *R*-matrix radii are possible. This is due to linear dependence arising between the various GTO basis sets used when bigger radii are employed. Replacing the GTO description of the continuum orbitals with other types of basis sets will enable us to use larger radii and describe collisions in which diffuse electronic states are involved.

The work performed on the UKRmol suite over the last 30 months is part of a larger project (UK-RAMP) whose aim is to develop a suite of high-quality computer codes to describe not only lepton scattering from molecules but also photoionisation by intense lasers. For this purpose, the inner region programs are being adapted to drive the atomic *R*-Matrix with Time-dependence (RMT) codes [88,89].

Finally we note that the codes are distributed as free-ware via the CCPForge repository. It is also possible to run them using the Quantemol-N expert system [90].

This work was performed as part of the UK-RAMP consortium which is an EPSRC funded HPC Software Development Project. Support from STFC, NAG Ltd and Quantemol Ltd is also gratefully acknowledged. We also thank the many users of the codes for their input into the software development project.

References

1. D.R. Bates, Phys. Rev. **78**, 492 (1950)
2. D.L. Huestis, S.W. Bougher, J.L. Fox, M. Galand, R.E. Johnson, J.I. Moses, J.C. Pickering, Space Sci. Rev. **139**, 63 (2008)
3. A. Luque, U. Ebert, W. Hundsdofer, Phys. Rev. Lett. **101**, 075005 (2008)
4. J.M. Ajello, D.E. Shemansky, W.R. Pryor, A.I. Stewart, K.E. Simmons, T. Majeed, J.H. Waite, G.R. Gladstone, D. Grodent, Icarus **152**, 151 (2001)
5. W.D. Geppert, M. Larsson, Mol. Phys. **106**, 2199 (2008)
6. A. Fuente, S. Garca-Burillo, A. Usero, M. Gerin, R. Neri, A. Faure, J. Le Bourlot, M. Gonzalez-Garca, J.R. Rizzo, T. Alonso-Albi, J. Tennyson, Astron. Astrophys. **492**, 675 (2008)
7. M. Roellig, N.P. Abel, T. Bell, F. Bensch, J. Black, G.J. Ferland, B. Jonkheid, I. Kamp, M.J. Kaufman, J. Le Bourlot, F. Le Petit, R. Meijerink, O. Morata, V. Ossenkopf, E. Roueff, G. Shaw, M. Spaans, A. Sternberg, J. Stutzki, W.-F. Thi, E.F. van Dishoeck, P.A.M. van Hoof, S. Viti, M.G. Wolfire, Astron. Astrophys. **467**, 187 (2007)
8. B. Boudaiffa, P. Cloutier, D. Hunting, M.A. Huels, L. Sanche, Science **287**, 1658 (2000)
9. I. Baccarelli, I. Bald, F.A. Gianturco, E. Illenberger, J. Kopyra, Phys. Rep. **508**, 1 (2011)
10. M.J. Brunger, S.J. Buckman, Phys. Rep. **357**, 215 (2002)
11. C. Winstead, V. McKoy, Adv. At. Mol. Phys. **43**, 111 (2000)
12. B.I. Schneider, T.N. Rescigno, Phys. Rev. A **37**, 3749 (1988)
13. T.N. Resigno, C.W. McCurdy, A.E. Orel, B.H. Lengsfeld III, The complex Kohn variational method, in *Computational Methods for Electron Molecule Collisions*, edited by W.M. Huo, F. Gianturco (Plenum Press, New York, 1995), pp. 1–44
14. K. Takatsuka, V. McKoy, Phys. Rev. A **24**, 2473 (1981)
15. K. Takatsuka, V. McKoy, Phys. Rev. A **30**, 1734 (1984)
16. C. Winstead, Q.Y. Sun, P.G. Hipes, M.A.P. Lima, V. McKoy, Aust. J. Phys. **45**, 325 (1992)
17. P.G. Burke, *R-Matrix Theory of Atomic Collisions: Application to Atomic, Molecular and Optical Processes*, (Springer, Berlin, 2011)
18. J. Tennyson, Phys. Rep. **491**, 29 (2010)
19. L.A. Morgan, J. Tennyson, C.J. Gillan, Comput. Phys. Commun. **114**, 120 (1998)
20. L.A. Morgan, C.J. Gillan, J. Tennyson, X. Chen, J. Phys. B At. Mol. Opt. Phys. **30**, 4087 (1997)
21. N.F. Lane, Rev. Mod. Phys. **52**, 29 (1980)

22. M. Tarana, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **41**, 205204 (2008)
23. K.A. Berrington, C.P. Ballance, *J. Phys. B At. Mol. Opt. Phys.* **35**, 2275 (2002)
24. J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **37**, 1061 (2004)
25. D.W. Norcross, N.T. Padial, *Phys. Rev. A* **25**, 226 (1982)
26. J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **29**, 6185 (1996)
27. K. Bartschat, E.T. Hudson, M.P. Scott, P.G. Burke, V.M. Burke, *J. Phys. B At. Mol. Opt. Phys.* **29**, 115 (1996)
28. J.D. Gorfinkiel, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **37**, L343 (2004)
29. J.D. Gorfinkiel, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **38**, 1607 (2005)
30. M.W. Schmidt, K. Ruedenberg, *J. Chem. Phys.* **71**, 3951 (1992)
31. C.J. Noble, P.G. Burke, S. Salvini, *J. Phys. B At. Mol. Phys.* **15**, 3779 (1982)
32. J. Almlöf, P.R. Taylor, in *Advanced Theories and Computational Approaches to the Electronic Structure of Molecules*, edited by C.E. Dykstra (Reidel, Dordrecht, 1984)
33. H.-J. Werner et al., *Molpro, version 2008.1, a package of ab initio programs* (2008), see <http://www.molpro.net>.
34. M. Jones, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **43**, 045101 (2010)
35. K.L. Baluja, P.G. Burke, L.A. Morgan, *Comput. Phys. Commun.* **27**, 299 (1982)
36. C.J. Noble, R.K. Nesbet, *Comput. Phys. Commun.* **33**, 399 (1984)
37. A.R. Barnett, *Comput. Phys. Commun.* **27**, 147 (1982)
38. J. Tennyson, C.J. Noble, *Comput. Phys. Commun.* **33**, 421 (1984)
39. D.T. Stibbe, J. Tennyson, *Comput. Phys. Commun.* **114**, 236 (1998)
40. N. Sanna, F.A. Gianturco, *Comput. Phys. Commun.* **114**, 142 (1998)
41. J. Franz, K.L. Baluja, R. Zhang, J. Tennyson, *Nucl. Instr. Meth. B* **266**, 419 (2008)
42. B.K. Sarpal, S.E. Branchett, J. Tennyson, L.A. Morgan, *J. Phys. B At. Mol. Opt. Phys.* **24**, 3685 (1991)
43. A.G. Harvey, F. Morales, O. Smirnova, to be published
44. A. Faure, J.D. Gorfinkiel, L.A. Morgan, J. Tennyson, *Comput. Phys. Commun.* **144**, 224 (2002)
45. B.M. Nestmann, S.D. Peyerimhoff, *J. Phys. B At. Mol. Opt. Phys.* **23**, L773 (1990)
46. A.G. Harvey, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **42**, 095101 (2009)
47. J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **21**, 805 (1988)
48. I. Rabadán, J. Tennyson, *Comput. Phys. Commun.* **114**, 129 (1998)
49. R. Zhang, K.L. Baluja, J. Franz, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **44**, 035203 (2011)
50. J.M. Carr, M. Plummer, J.D. Gorfinkiel, *Coding standards for the ukrmol-in project*, Technical report, STFC CSED and The Open University, 2010
51. S.J. Clark, M.D. Segall, C.J. Pickard, M.J. Probert, P.J. Hasnip, K. Refson, M.C. Payne, *Z. Kristallogr.* **220**, 567 (2004)
52. A.G. Sunderland, C.J. Noble, V.M. Burke, P.G. Burke, *Comput. Phys. Commun.* **145**, 311 (2002), see also <http://www.hector.ac.uk/cse/distributedcse/reports/prmat/>
53. ISO/IEC 1539-1:1997 (base language) and ISO/IEC TR 15581:2001 (extension concerning the use of allocatable arrays in derived types), at <http://www.iso.org>
54. J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **29**, 1817 (1996)
55. J. Tennyson, *Comput. Phys. Commun.* **100**, 26 (1997)
56. A. Stathopoulos, C.F. Fischer, *Comput. Phys. Commun.* **79**, 268 (1994)
57. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenn, D. Sorensen, *LAPACK Users' Guide*, 3rd edn. (SIAM, Philadelphia, 1999), <http://www.netlib.org/lapack>
58. R.B. Lehoucq, D.C. Sorensen, Chao Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods* (SIAM, 1998)
59. R. Zhang, P.G. Galiatsatos, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **44**, 195203 (2011)
60. G.H. Golub, C.F. van Loan, *Matrix Computations* (JHU Press, 1996)
61. V. Hernandez, J.E. Roman, V. Vidal, *ACM Trans. Math. Softw.* **31**, 351 (2005)
62. S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, *PETSc Web page* (2011), <http://www.mcs.anl.gov/petsc>
63. J. Tennyson, *J. Phys. B At. Mol. Phys.* **19**, 4255 (1986)
64. G. Danby, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **23**, 1005 (1990)
65. G. Danby, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.*, Erratum **23**, 2471 (1990)
66. K.L. Baluja, R. Zhang, J. Franz, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **40**, 3515 (2007)
67. A. Faure, J.D. Gorfinkiel, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **37**, 801 (2004)
68. A. Dora, L. Bryjko, T. van Mourik, J. Tennyson, *J. Chem. Phys.* **130**, 164307 (2009)
69. S. Caprasecca, J.D. Gorfinkiel, D. Bouchiha, L.G. Caron, *J. Phys. B* **42**, 095205 (2009)
70. H.N. Varambhia, A. Faure, K. Graupner, T.A. Field, J. Tennyson, *Mon. Not. R. Astron. Soc.* **403**, 1409 (2010)
71. V. Kokouline, A. Faure, J. Tennyson, C.H. Greene, *Mon. Not. R. Astron. Soc.* **405**, 1195 (2010)
72. I. Rabadán, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **32**, 4753 (1999)
73. B.K. Sarpal, J. Tennyson, L.A. Morgan, *J. Phys. B At. Mol. Opt. Phys.* **27**, 5943 (1994)
74. I.F. Schneider, I. Rabadán, L. Carata, J. Tennyson, L.H. Andersen, A. Suzor-Weiner, *J. Phys. B At. Mol. Opt. Phys.* **33**, 4849 (2000)
75. K. Chakrabarti, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **40**, 2135 (2007)
76. K. Chakrabarti, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **44**, 055203 (2011)
77. D. Madden, J. Tennyson, R. Zhang, *J. Phys.: Conf. Ser.* **300**, 012017 (2011)
78. G. Halmová, J.D. Gorfinkiel, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **39**, 2849 (2006)
79. L. Bryjko, T. van Mourik, A. Dora, J. Tennyson, *J. Phys. B At. Mol. Opt. Phys.* **43**, 235203 (2010)
80. S. Harrison, J. Tennyson, submitted to *J. Phys. B At. Mol. Opt. Phys.*

81. Zdeněk Mašín, J.D. Gorfinkiel, J. Chem. Phys. **135**, 144308 (2011)
82. C.S. Trevisan, J. Tennyson, J. Phys. B At. Mol. Opt. Phys. **34**, 2935 (2001)
83. C.S. Trevisan, J. Tennyson, Plasma Phys. Control. Fusion **44**, 1263 (2002)
84. J.D. Gorfinkiel, L.A. Morgan, J. Tennyson, J. Phys. B At. Mol. Opt. Phys. **35**, 543 (2002)
85. G. Halmová, J. Tennyson, Phys. Rev. Lett. **100**, 213202 (2008)
86. G. Halmová, J.D. Gorfinkiel, J. Tennyson, J. Phys. B At. Mol. Opt. Phys. **41**, 155201 (2008)
87. R. Zhang, A. Faure, J. Tennyson, Phys. Scr. **80**, 015301 (2009)
88. M.A. Lysaght, L.R. Moore, L.A.A. Nikolopoulos, J.S. Parker, H.W. van der Hart, K.T. Taylor, *Quantum Dynamic Imaging: Theoretical and Numerical Methods* (Springer, 2011)
89. L.R. Moore, M.A. Lysaght, L.A.A. Nikolopoulos, J.S. Parker, H.W. van der Hart, K.T. Taylor, J. Mod. Opt. **58**, 1132 (2011)
90. J. Tennyson, D.B. Brown, J.J. Munro, I. Rozum, H.N. Varambhia, N. Vinci, J. Phys.: Conf. Ser. **86**, 012001 (2007)