Письма в ЭЧАЯ. 2008. Т. 5, № 3(145). С. 273–282

E-COLLISIONS USING E-SCIENCE

N. S. Scott [a], *V. Faro-Maza* [a], *M. P. Scott* [b], *T. Harmer* [c], *J.-M. Chesneaux* [d], *C. Denis* [d], *F. Jézéquel* [d]

[a] School of Electronics, Electrical Engineering and Computer Science, The Queen's University of Belfast, Belfast, UK

[b] School of Mathematics and Physics, The Queen's University of Belfast, Belfast, UK

[c] Belfast e-Science Centre, The Queen's University of Belfast, Belfast, UK

[d] Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie — Paris 6, France

We describe a computational science research programme primarily aimed at engineering numerically robust software that can exploit high performance on distributed computers in the study of electron collisions with atoms and ions. In particular, we describe the development of 2DRMP-G, a Grid aware two-dimensional R-matrix propagator, and its numerical validation using CADNA, a software tool based on discrete stochastic arithmetic.

Описана программа вычислительных исследований, первоначально предназначенная для создания численно надежного программного обеспечения, которое может использовать высокую производительность распределенных компьютеров для изучения столкновений электронов с атомами и ионами. В частности, описано создание программы 2DRMP-G, являющейся грид-приложением для вычисления двумерного R-матричного пропагатора, и численная проверка ее достоверности с помощью программы CANDA, программная реализация которой основана на дискретной стохастической арифметике.

PACS: 02.60.-X; 02.90.+P

INTRODUCTION

For over four hundred years, from the Scientific Revolution at the dawn of the 17th century to the end of the Second World War, modern science progressed through a rich interplay between theory and experiment.

This changed dramatically following the development of the digital computer in the 1940s. Computational science was born and, after cutting its teeth on the ballistics and nuclear weapons problems of World War II, it has now emerged as a mature, powerful and indispensable methodology in scientific research — providing a third way of doing science and complementing the traditional approaches of theory and experiment.

Computational science is an eclectic mix of mathematics, computer science and an applied discipline such as physics. It is concerned with the complete computational process. It not only seeks to advance science through the use of high-performance and distributed computers but also to advance the state of the art in computer systems by studying scientific applications that expose the limitations and weaknesses of these systems.

In this paper we describe a computational science research programme that is primarily aimed at engineering numerically robust software that can exploit high-performance and distributed computers to study electron collisions with atoms and ions. A further aim is to provide a realistic test bed application of substance that can inform and stimulate the further development of modern computer systems including Grid technology and reconfigurable high-performance computers. We begin by briefly introducing the physics of electron collisions (e-collisions). Next we sketch an apposite mathematical model that is based on R-matrix theory [1]. A computational realization of this model that can exploit Grid technology, 2DRMP-G, is presented in Sec. 3 (e-science). Finally, we describe the use of the CADNA library [2], a software tool based on stochastic discrete arithmetic that is used to validate the stability of numerical software.

1. PHYSICS: E-COLLISIONS

Electron collisions with atoms and ions have been the subject of international interest for many years. Data from these processes are of importance in the analysis of physical phenomena in many scientific and technological areas including aeronomy, astrophysics, biomedicine, gaseous electronics, surface physics, industrial plasmas, environmental, fusion, semiconductor and other technologies [3].

Despite the importance of these applications, relatively little accurate cross-section data is known for many of the processes involved. For example, no accurate cross-section data, involving high lying excited states, have been calculated for electron collisions with the simplest atomic target, hydrogen. Accordingly, the focus of this paper is on electron impact excitation of H-like atoms and ions at these, so-called, intermediate scattering energies.

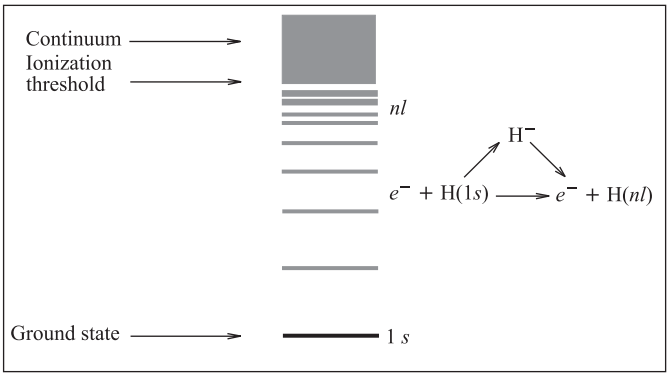

Fig. 1. Electron impact excitation. Here a free electron collides with an H-like atom or ion in its ground state resulting in an excited H-like atom and a free electron. This final state can be reached directly or via an intermediate resonance state

This process is illustrated in Fig. 1. The process can occur directly or via an intermediate resonance state, i.e., a quasi-bound state with a long lifetime. Resonance analysis is important in elucidating the collision process and in determining sensitive parameters against which theoretical computations and experimental results may be compared. Modelling this process is

computationally demanding because account must be taken of the infinite number of continuum states of the ionized target and of the infinite number of target bound states lying below the ionization threshold.

2. MATHEMATICAL MODEL: R-MATRIX THEORY

R-matrix theory was first introduced in nuclear physics by Wigner [5,6]. Around the 1960s it was realized that this approach could also be used in atomic and molecular physics [7]. Since then the R-matrix method has proved to be a remarkably stable, robust and efficient technique for solving the close-coupling equations that arise in electron collision theory [1]. We begin by sketching general R-matrix theory for $(N+1)$-electron targets.

2.1. General R-Matrix Theory. General R-matrix theory starts by dividing the configuration space describing the collision process into two regions by a sphere of radius $r = a$, where a is chosen so that the charge distribution of the target atom or ion is contained within the sphere. In the internal region ($r \leqslant a$) exchange and correlation effects between the scattering electron and the target electrons must be included, whereas in the external region such effects can be neglected thereby considerably simplifying the problem.

In the internal region the $(N+1)$-electron wave function at energy E is expanded in terms of an energy independent basis set, ψ_k, as

$$\Psi_E = \sum_k A_{Ek}\psi_k. \tag{1}$$

The basis states, ψ_k, are themselves expanded in terms of a complete set of numerical orbitals, u_{ij}, constructed to describe the radial motion of the scattered electron. The expansion coefficients of the u_{ij} set can be determined by diagonalizing the following Hamiltonian matrix:

$$(\psi_i|H_{N+1}|\psi_j) = E_k^{N+1}\delta_{ij}, \tag{2}$$

where H_{N+1} is the $(N+1)$-electron Hamiltonian operator. This in turn allows the construction of the R-matrix,

$$R_{ij} = \frac{1}{2a}\sum_k \frac{\omega_{ik}(a)\omega_{jk}(a)}{E_k^{N+1} - E}, \tag{3}$$

at $r = a$, where the amplitudes $\omega_{ik}(a)$ and the poles E_k^{N+1} of the R-matrix are obtained directly from the eigenvectors and eigenvalues of Eq. (2).

In the outer region the equations reduce to coupled second-order ordinary differential equations. Using a technique such as 1D propagation, implemented in the FARM package [4], these equations can be integrated outwards, subject to the R-matrix boundary conditions at $r = a$, and fitted to an asymptotic expansion. This determines the K-matrix from which the scattering observables including inelastic cross sections and resonance positions and widths can be derived. While the R-matrix is determined by a single diagonalization in the inner region, the coupled equations in the outer region must be solved for each scattering energy of interest.

It can be shown that dimension of the Hamiltonian matrix, n, within the internal region is proportional to a^2. Since matrix diagonalization is an n^3 process the size of the inner-region computation rapidly increases when transitions to higher level states are computed.

For example, in electron scattering by atomic hydrogen a radius of $a = 60$ a.u. is needed to envelop the $n = 4$ target states, while a radius of 360 a.u. is required to envelop the $n = 10$ target states. This results in an increase in diagonalization time by a factor of approximately 46,000. Solution of the corresponding dense Hamiltonian matrices, which are typically of the order of 50,000 to 100,000, places considerable demands on both computer time and numerical robustness.

In the following subsection we outline an alternative approach that is designed for scattering from H-like systems. It has the advantage of extending the boundary radius of the inner region far beyond that which is possible using the traditional «one-sector» technique.

2.2. 2D R-Matrix Theory. As in general R-matrix theory the two-electron configuration space (r_1, r_2) is divided into two regions by a sphere of radius a centered on the target nucleus [8]. However, in the 2D variant the inner region is further divided into subregions as illustrated in Fig. 2.

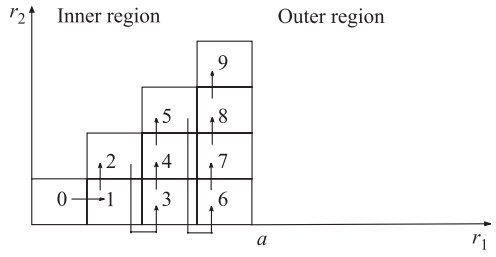

Fig. 2. Subdivision of the inner-region configuration space (r_1, r_2) into a set of connected subregions labelled $0 \ldots 9$

Within each subregion energy-independent R-matrix basis states, $\theta_k^{LS\pi}(\mathbf{r}_1, \mathbf{r}_2)$, are expanded in terms of one-electron basis functions, ν_{ij}, whose radial forms are solutions of the Schrödinger equation. The expansion coefficients of the ν_{ij} set are obtained by diagonalizing the corresponding two-electron Hamiltonian matrix. The expansion coefficients and the radial basis functions are then used to construct surface amplitudes, $\omega_{inl_1l_2k}$, associated with each subregion edge $i \in \{1, 2, 3, 4\}$.

For each incident electron energy a set of local R-matrices (R_{ji}) can be constructed from the surface amplitudes as follows:

$$(R_{ji})_{n'l_1'l_2'nl_1l_2} = \frac{1}{2a_i} \sum_k \frac{\omega_{jn'l_1'l_2'k}\omega_{inl_1l_2k}}{E_k - E}, \quad j, i \in \{1, 2, 3, 4\}. \tag{4}$$

Here a_i is the radius of the ith edge; E is the total energy of the two-electron system and E_k are the eigenenergies obtained by diagonalizing the two-electron Hamiltonian in the subregion. By using the local R-matrices, the R-matrix on the boundary of the innermost subregion can be propagated across all subregions, working systematically from the r_1-axis at the bottom of each strip to its diagonal as illustrated in Fig. 2, to yield the global R-matrix, $\Re$, on the boundary of the inner and outer region ($r_1 = a$).

Finally, the global R-matrix $\Re$ is transformed onto an appropriate basis for use in the outer region. The resulting outer-region equations are identical in form to those described in Subsec. 2.1.

3. COMPUTATIONAL MODEL: E-SCIENCE

The 2D propagator model described in the previous section has been implemented as the suite of seven programs depicted in Fig. 3. These programs have been designed to operate on serial machines and to exploit distributed memory and shared memory parallelism found on tightly coupled high-performance clusters.

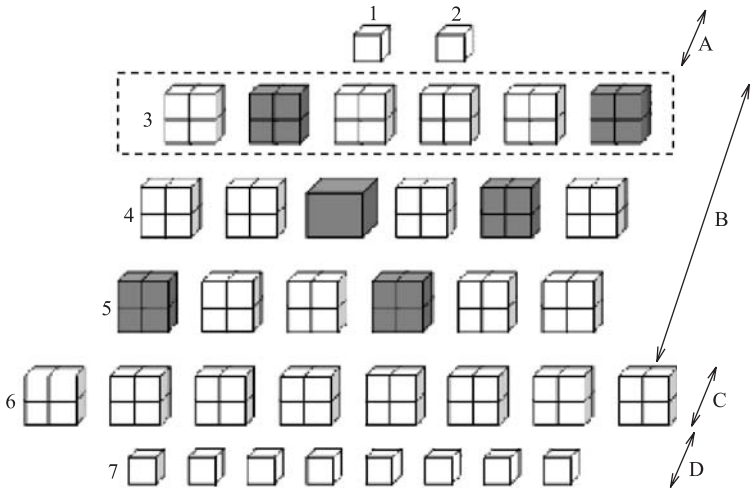

Fig. 3. The 2DRMP package. Blocks A and B are independent of the collision energy and need only be performed once, while blocks C and D are dependent on the collision energy and must be repeated hundreds of times

3.1. 2DRMP. We begin by sketching the 2DRMP suit. Each of the seven programs belongs to one of the four functional blocks shown in Fig. 3: A, B, C or D. The blocks must be executed sequentially and communication between programs is through files.

Block A contains two independent programs that are not computationally intensive. Program 1 constructs the atomic basis functions used in the transformation of the global R-matrix, $\Re$, while program 2 computes radial integrals to be used in the construction of the Hamiltonian matrix in off-diagonal subregions.

In block B, program 3 constructs a subregion Hamiltonian matrix, program 4 diagonalizes the matrix, and program 5 constructs the corresponding surface amplitudes from the matrix's eigenvalues and eigenvectors. Each column in this block corresponds to an independent subregion.

Block C uses program 6 to propagate the global R-matrix, $\Re$, across all the subregions of the inner region. Each element in this block corresponds to a series of propagations, one for each scattering energy.

Block D corresponds to an outer-region program such as FARM. Again each element in this block corresponds to a range of scattering energies.

Programs 3, 4, 5 and 6 can execute in serial or parallel. For example, program 3 can construct a matrix in serial or in parallel using either MPI or OpenMP. Programs 4, 5 and 6 have a similar capability. On a tightly coupled supercomputer, such as HPCx [9], parallelism tends to be horizontal with, for example, many matrices being computed simultaneously

(controlled by MPI) and each being spread over several processors (controlled by either MPI or OpenMP). This is illustrated by the dotted rectangle in Fig. 3 where each matrix construction is itself spread across a 2×2 grid of processors.

3.2. 2DRMP-G. In common with many traditional scientific applications 2DRMP does not have any real-time constraints in its execution or urgency in obtaining results. There is, therefore, no pressing need to use the fastest available computational resource. Any collection of computational resources that can deliver results in a reasonable time is acceptable. A Grid infrastructure provides an environment for the cooperative use of distributed computational and storage resources and Grid middleware enables the secure use of remote resources and the transport of data within the infrastructure. On a Grid of heterogeneous computers, such as the UK L2 Grid [10], parallelism within block B can extend both vertically and horizontally. This is illustrated by the darkened elements in Fig. 3 where each element is computed at a different site each with potentially different system architectures. For example, one of the matrix diagonalizations (program 4) might be performed on a commodity PC using LAPACK, while the second could use ScaLAPACK, or a multi-threaded version of LAPACK, across a 2×2 grid on tightly coupled cluster.

In this subsection we describe a Grid aware version of 2DRMP aimed specifically at computing resonance parameters in electron scattering by H-like atoms and ions at intermediate energies. 2DRMP-G is composed of the three elements depicted in Fig. 4.

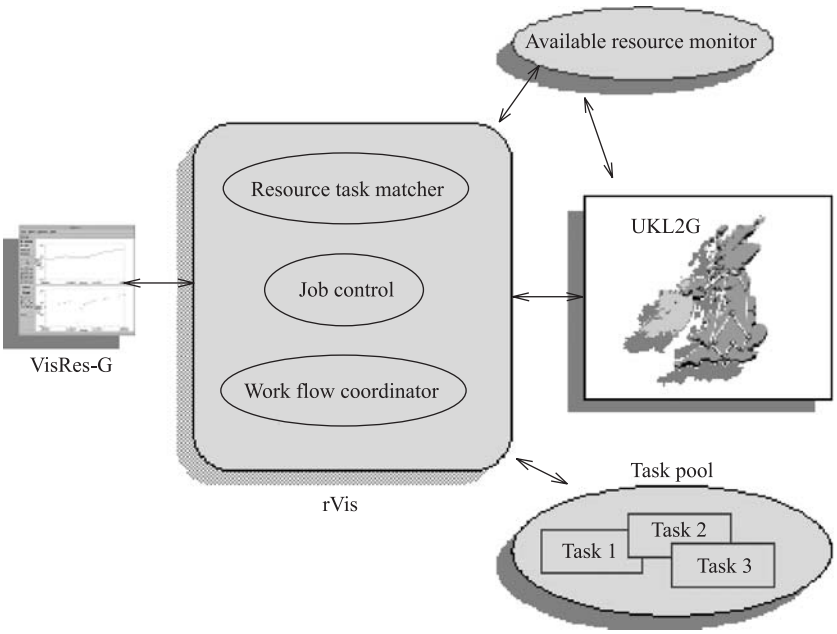

Fig. 4. The 2DRMP-G Grid architecture composed of the three components VisRes-G, rVis and a task pool of tasks from 2DRMP

1. VisRes-G: A GUI tool for computational steering [11].
2. rVis: The Grid middleware tool used to control the Grid computations.

3. 2DRMP: The 2DRMP suite of programs described in Subsec. 3.1 that contributes tasks to the task pool.

The Grid middleware tool rVis is written in Java and built on top of Globus 2. It uses the concept of a task pool to define the processing that is possible at any stage during execution. A work flow coordinator controls execution by scheduling tasks to the task pool. The work flow dependencies of the application are defined in a simple XML database. A resource-task allocation component attempts to match tasks that are in the task pool with resources that are currently available in the computation Grid. The job control component is used to control the execution of a task on a computational resource. Using Globus 2 its role is to package and transport the data that a task requires, to start job execution, to retrieve results, to detect and reschedule a task if execution fails and to log job submission details.

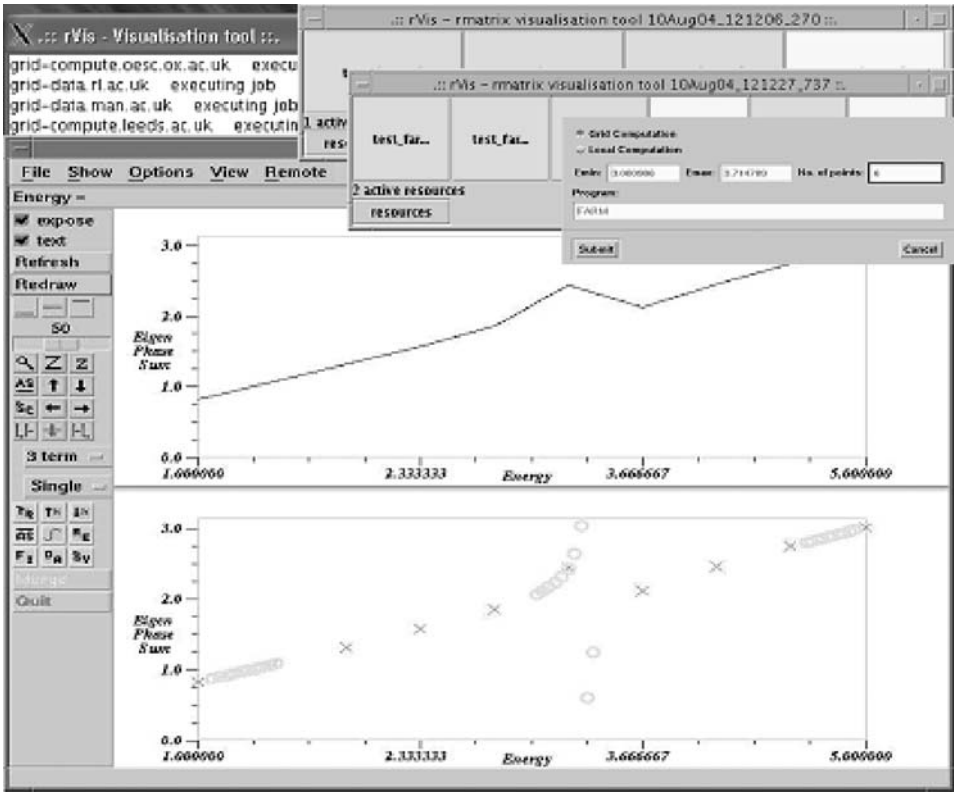

Fig. 5. The initial eigenphase data returned to VisRes-G by 2DRMP-G is illustrated by the crosses. By clicking and dragging across the lower graph a new collection of Grid computations is initiated, collected and redisplayed in the form of the circles

VisRes-G, as illustrated by Fig. 5, is a comprehensive visual tool that facilitates the graphical display, manipulation and analysis of resonance data computed over a Grid.

Resonances are revealed when the eigenphase sum, $\delta(E)$, rises rapidly by approximately π radians over a small range of collision energies. Their positions are unknown at the outset of

the computation and the initally chosen scattering energies often provide an insufficient number of data points within the vicinity of the resonance to enable its characteristics accurately to be determined. Hence the need for computational steering arises.

VisRes-G is initially used to select an initial range of collision energies and to instigate the computation of the tasks depicted in Fig. 3. VisRes-G uses rVis to control the computation of tasks across the Grid. The output from the final stage of 2DRMP is automatically returned to VisRes-G and displayed on the user's workstation as illustrated by the crosses in Fig. 5. By clicking and dragging across the graph the user selects collections of points where extra data may be required, thereby steering the computation towards potential resonances. These tasks are added to the task pool, each task corresponding to the computation of an eigenphase at a distinct energy. The resulting eigenphase data is returned to VisRes-G and merged with the original data. For example, the data from three new collections of points are displayed as circles in Fig. 5. Further data can be generated in a similar fashion until the user is satisfied that all the necessary data has been computed. The position and width of the resulting resonances can then be analyzed using tool-box functions.

4. NUMERICAL VALIDATION: CADNA

A novel aspect of this work is the rigorous investigation of the numerical validation of large-scale distributed scientific computation. This is a topic of considerable importance, but one that has received relatively little practical attention in the computational science community. It is well known that the floating point arithmetic commonly used in scientific computing only approximates exact arithmetic. In consequence, each arithmetic statement generates a round-off error. It is not uncommon to find that the same code, using the same data, produces different results when executed on different platforms. This is particularly the case when different levels of compiler optimization are used. Sometimes the result is not just inaccurate but totally wrong. In a Grid environment, where the overall computation may involve contributions from many heterogeneous platforms, controlled and rigorous numerical validation is essential.

Using Discrete Stochastic Arithmetic [12, 13] the CADNA (Control of Accuracy and Debugging for Numerical Applications) library [2] is a tool designed to estimate precisely the computing error in computer generated results, i.e., to estimate the number of common significant figures between the computed result and the exact result.

The basic idea is to perform the same computation several times propagating the round-off error differently each time. The computer's deterministic arithmetic is replaced by a stochastic arithmetic where each elementary operation is performed N times before the next instruction is executed. For each operation N samples are obtained. The mean value and standard deviation characterize the corresponding stochastic number. The value of a stochastic number is the mean value of the different samples. The number of exact significant digits in the number is estimated using the mean value and the standard deviation. If all the samples are zero or if the number of exact significant digits is less than one the number is defined as a computational zero denoted by @.0. This means that a computational zero is either a mathematical zero or a number without any significance.

The CADNA Fortran implementation is a set of data types, functions and subroutines that may be easily incorporated into any Fortran program. In essence, Fortran types are

simply replaced by the corresponding stochastic types. The stochastic numbers are N-tuples containing the perturbed floating point values. Arithmetic operators, logical operators and intrinsic functions have been overloaded so that when an operator is used the operands are N-tuples and the returned result is an N-tuple. The library, therefore, enables a scientific code to be executed using random arithmetic without having to make major changes to the original code. During execution when a numerical anomaly is detected a message is written to a special file. At the end of the run the user must consult the file, analyze the contents and determine if code changes are required. A nice feature is that intermediate and final results are output only to the exact number of digits.

In this project we are using CADNA systematically throughout 2DRMP to estimate the computing error of intermediate and final results. CADNA has been integrated into a selection of subroutines in sequential parts of the 2DRMP package to investigate numerical instability. In particular, the algorithm to compute the two-dimensional radial integrals (Slater integrals) has been modified to remove the numerical instabilities identified by CADNA [14]. We have also investigated the use of CADNA to control dynamically the optimum integration step used in the computation of the Slater integrals; i.e., the step for which the global error, consisting of both the truncation error and the round-off error, is minimal [15]. This has enabled the computation of benchmark results for a range of Slater integrals accurate to 10 exact significant digits [14].

Work is also in progress to implement CADNA within the complete 2DRMP package. Here we intend to employ MPI_CADNA, a variant of CADNA, to investigate the numerical stability of 2DRMP code across a large collection of processors in Grid'5000 [16].

CONCLUDING REMARKS

In this paper we have described a computational science research programme aimed at exploiting high-performance and distributed computers to enable virtual experiments to be performed where electrons collide with H-like atoms and ions. This work resulted in the development of a Grid aware two-dimensional R-matrix propagator, 2DRMP-G and an associated prototype visualization and computational steering tool, VisRes-G. VisRes-G contains software to monitor the resources that are available in the Grid and dynamically to allocate available tasks to available resources. A major benefit of VisRes-G is that it enables the user to focus solely on steering the physics without having to be aware of the computational resources being used.

Like all scientific software, 2DRMP-G has approximations woven into its fabric at each level: in its mathematical model, its computational model and its computer implementation. One important source of error that is difficult to understand and manage is the propagation of round-off error that originates from the use of finite precision arithmetic. This problem is exacerbated in a supercomputing environment where trillions of floating-point operations may be performed per second and in a Grid environment, where the overall computation may involve contributions from many heterogeneous platforms. Controlled and rigorous numerical verification is recommended to give confidence that the computed results are acceptable. We have advocated the use of CADNA to provide such a software «health check».

Acknowledgements. The authors are grateful to the UK EPSRC for their support through the UK e-Science Programme and grants GR/M01784/01 and GR/R89073/01. V.F.M. gratefully acknowledges the receipt of an ESF/Queen's University postgraduate studentship.

REFERENCES

1. Atomic and Molecular Processes and R-matrix Approach / Eds. P. G. Burke, K. A. Berrington. Bristol: IOP Publ., 1993.
2. CADNA website - http://www.lip6.fr/cadna
3. *Becker K. H. et al.* // Workshop on Electron-Driven Processes: Scientific Challenges and Technological Opportunities. http://attila.stevens-tech.edu/physics/People/Faculty/Becker/EDP/ EDP-FinalReport.pdf
4. *Burke V. M., Noble C. J.* // Comp. Phys. Commun. 1995. V. 85. P. 471.
5. *Wigner E. P.* // Phys. Rev. 1946. V. 70. P. 15.
6. *Wigner E. P.* // Ibid. P. 606.
7. *Burke P. G.* // Adv. At. Mol. Phys. 1968. V. 4. P. 173.
8. *Burke P. G., Noble C. J., Scott M. P.* // Proc. Roy. Soc. A. 1987. V. 410. P. 287.
9. HPCx, UK National Supercomputing Centre. http://www.hpcx.ac.uk/
10. *Alan R. et al.* UK e-Science All Hands Meeting. 2003. P. 556; http://www.nesc.ac.uk/events/ahm2003/AHMCD/pdf/125.pdf
11. *Faro-Mazo V. et al.* UK e-Science All Hands Meeting. 2004. P. 1007; http://www.allhands.org.uk/2004/proceedings/papers/134.pdf
12. *Vignes J.* // Math. Comp. Simulation. 1993. V. 35. P. 233.
13. *Vignes J.* // Num. Algo. 2004. V. 37. P. 377.
14. *Scott N. S. et al.* // Comp. Phys. Commun. 2007 (in press).
15. *Chesneaux J.-M. et al.* Scan2006, Duisburg, Germany.
16. Grid'5000. http://www.lri.fr/ fci/Grid5000/