Report on eCSE project "Performance enhancement in R-matrix with time-dependence (RMT) codes in preparation for application to circular polarised light fields"

J S Parker (QUB), Martin Plummer (STFC), H W van der Hart (QUB)

Version 1.0, September 29, 2015

Abstract

We describe the results of a rewrite of the inner-region communication algorithms of the RMT code, and the development of a new load balancing algorithm. The new design yielded a speed-up of 500% in the test programs, 300% in certain production runs, and reduced the amount of RAM used during initialisation by one or more orders of magnitude. The improved performance will enable treatment of multi-electron atom-laser interactions in laser radiation with arbitrary polarisation, and has demonstrated immediate benefits in studies of harmonic generation in singly-ionized Neon and attosecond transient absorption spectroscopy of neon.

1. Introduction

The R-matrix with time dependence (RMT) method is a new ab-initio method for solving the time-dependent Schroedinger Equation (TDSE) for multi-electron atomic and molecular systems in intense short laser pulses. Although several other time-dependent R-Matrix methods have been introduced in recent years, RMT has demonstrated significant improvements in efficiency and accuracy, primarily because the finite-difference (FD) techniques used to model the few-electron wavefunction far from the atomic core allow highly accurate and efficient explicit Arnoldi propagators to be used. The RMT [2, 5] codes adopt the standard R-matrix approach of separating the problem into an inner-region problem and an outer-region problem in configuration space. RMT merges an Outer Region FD model with a B-Spline R-Matrix basis set [6] for the multi-electron Inner Region, using initial data from the field-free R-matrix codes [1, 4]. The difficult problem of merging a basis set model with a spatially adjacent FD model while maintaining the unitarity of the time-propagator has been a long-standing barrier to progress in this field. RMT is based on the solution to this problem, first published in 2008 by Nikolopoulos, Parker and Taylor [3]. We have now shown that this method is both computationally stable and highly efficient.

The Fortran-95-based RMT codes have been developed for a wide range of platforms, including HECToR, ARCHER, and a local cluster at QUB. Early applications of the RMT codes were carried out for small-scale calculations (2 residual-ion states for Ne,

and a single residual-ion state for He). In the last two years, however, we have started to apply the codes to large-scale problems. These studies include the influence of atomic structure on IR-assisted photoionization in Ne+, and double ionization processes in He. These calculations require extensive configuration-interaction (CI) descriptions within each angular momentum. In typical problems, these CI expansions increase by 1 (Ne^+) or 2 (He) orders of magnitude.

As noted, the RMT codes adopt the standard R-matrix approach of separating the problem into an inner-region problem and an outer-region problem in configuration space. In the outer-region, the time-dependent wave function is described on a multi-channel radial grid and is parallelised efficiently with MPI over grids with many 100s of channels. In the inner region, the wavefunction is described in terms of the CI expansion within each symmetry component.

In this report we discuss the rewrite of the RMT inner-region algorithms. The new code uses less RAM, and allows us to use Hamiltonians that are an order of magnitude larger. In Phase 1 of the project a new load balancing procedure was developed for assigning cores to the inner region matrix-vector multiplications, which turned out to be essential in the new class of physical problems this project was designed to address. The new procedure produces a factor of 5 speed-up of the inner-region numerical integration, and consequently a 5-fold speed-up of the global integration. In phases 2 and 3, we optimized the initial inner-region communication algorithms. The original RMT code used a single MPI master task to read a Hamiltonian block matrix from the hard drive, and then distribute portions of it to other inner-region cores. The larger Hamiltonians addressed by the new RMT code are potentially 5 gigabytes (or more) in size, making this approach unfeasible. The new code simply reads the blocks incrementally, distributes the data in small blocks, and repeats until end-of-file is reached.

2. Phase 1: Load balancing in inner region matrix-vector algorithms

As an example problem, we have integrated the Schroedinger equation for singly-ionized Neon in intense near-UV (390 nm) radiation. In the RMT approach, the integration is performed by radically different numerical methods in the inner- and outer-regions of configuration space, as described above. Finite-difference methods are used to integrate the equation at distances far from the atom (the outer-region). In the inner region, an R-Matrix basis set represents the atom. This basis set decomposition gives rise to large dense Hamiltonian matrices, with different Hamiltonian matrices for each of the 23 symmetries in the inner-region decomposition. Large-scale matrix-vector multiplications are used to integrate the Schroedinger equation in this region.

The outermost parallelization layer in the inner-region RMT codes is to assign a separate set of MPI tasks to each symmetry. Each symmetry is associated with several matrix-vector multiplications. This computation is performed with anywhere from one to 15 cores per symmetry. Each symmetry communicates with neighboring symmetries, which ultimately hampers parallelization. Since each symmetry has a different CI expansion length, the problem here is to find the right number of cores to allocate to each symmetry in order to maximise performance, and to rewrite the code in order to perform the new allocation.

The workload of each symmetry is to a good approximation just the product of rows and columns in the matrix multiplication, (which is proportional to the number of floating point operations performed). The workload is expressed as a function, and the code attempts to allocate a number of cores to each symmetry in proportion to its workload. The workload function, $W(i)$, for symmetry i is:

$$W(i) = [\mathrm{Rows}(i-1) * \mathrm{Columns}(i) + \mathrm{Rows}(i+1) * \mathrm{Columns}(i)]^p,$$

where p is an arbitrary parameter. The parameter p, is introduced to address the fact that communications overhead makes the $p = 1$ version of $W(i)$ sub-optimal. The optimal

value of p turned out to be about 0.9. Values of p near 0 would imply that the work performed per symmetry is uniform. In this limit, ($p = 0$), allocation of cores would revert to the original strategy of assigning a constant number of cores to each symmetry. A minimum of 1 core is allocated per symmetry.

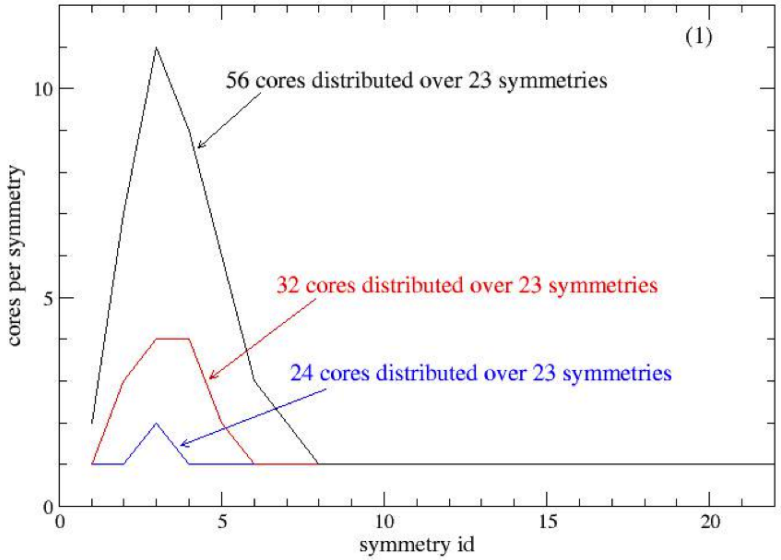

Figure 1: Distribution of cores over the different symmetries using the new workload function for the Ne+ test case. The lowest symmetry ids tend to need significantly more resources than the higher symmetry ids.

Figure 1 shows the distribution of cores generated by the new allocation rules. The old allocation rules resulted in a nearly uniform distribution of cores among the symmetries. In other words $23 \times N$ cores would be allocated in such a way that each symmetry would be assigned exactly N cores. Figure 1 shows that the new rules guarantee at least one core per symmetry, but favour the larger workloads in symmetries 2 through 6 according the formula described above. The CI expansion length in symmetries 3 and 4 is about a factor 6 larger than the expansion length in symmetries with id > 15.

Figure 2 above shows the speed-up obtained by the new strategy. An improvement

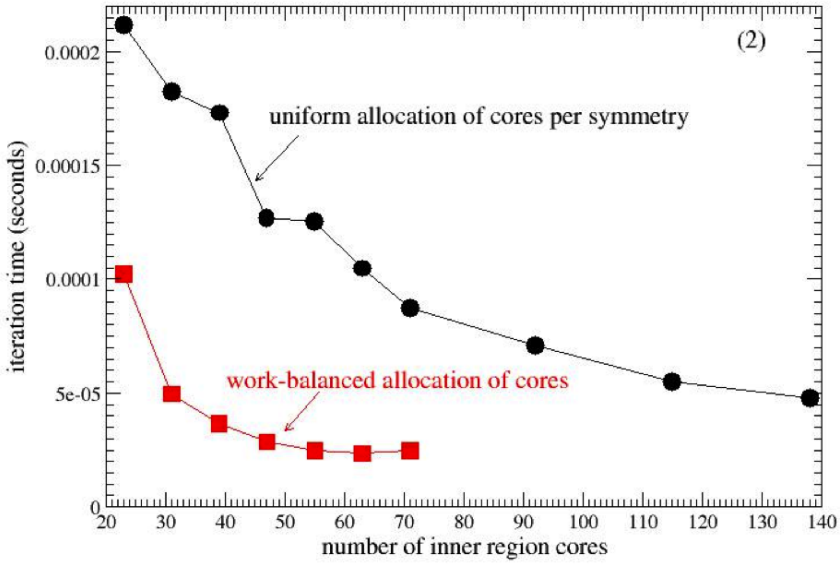

Figure 2: Iteration time as a function of total number of cores assigned to the inner region using a uniform allocation (black line with circles) and using the work-balanced allocation (red line with squares) for a small Ne+ test case.

by a factor of 5 is observed when 56 cores are used by the inner region integration. Beyond 56 cores, little improvement is obtained, because communication overhead starts to dominate. The old strategy shows gradual improvement as more cores are added, but even the use of 138 cores produces an executable that runs at half the speed of the new version with 56 cores.

Figure 3 shows the speed-up obtained during production runs of a heavily used RMT integration on ARCHER. In this case only 10 symmetries are used, which presents the load optimizer with fewer opportunities for optimization. Nevertheless the speed-up at 480 cores is a factor of 3. Any additional symmetry to be added would add relatively little work to the overall work required. Such additions would add significantly to the iteration time for a uniform allocation, but much less to the iteration time for a work-balanced core allocation.

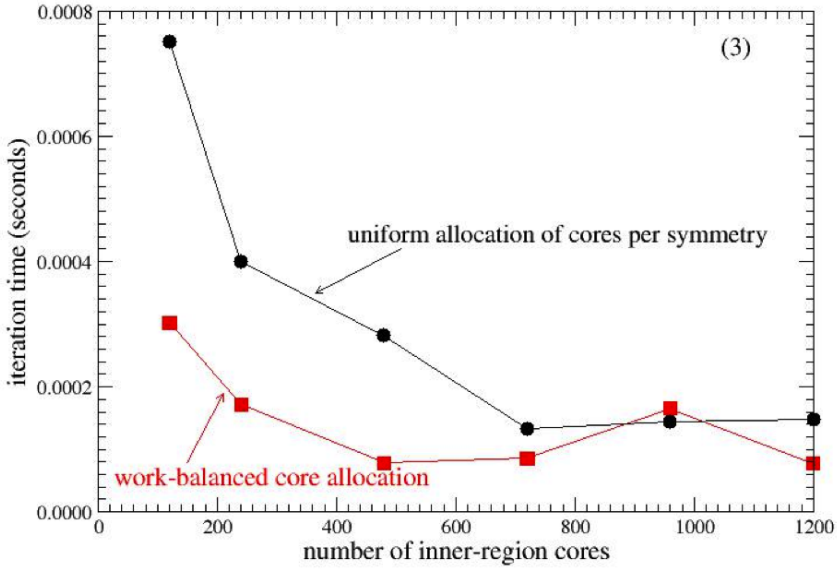

Figure 3: Iteration time as a function of total number of cores assigned to the inner region using a uniform allocation (black line with circles) and using the work-balanced allocation (red line with squares) for a large Ne+ test case using a reduced number of symmetries.

3. Phases 2 and 3: redesign of the communication during the initialization

Phases 2 and 3 of the project were a redesign of the initialization process for the Hamiltonian matrices used by the inner region R-Matrix integration (Phase 2), with subsequent parallelization of the read-in of bulk input data comprising field-free Hamiltonian eigenvalues and radial basis boundary values, dense dipole blocks and detailed channel data (Phase 3). The immediate goal of the redesign is to enable efficient set-up of the calculations of IR-assisted photoionization and double ionization.

The size of the off-diagonal blocks of the Hamiltonian in these calculations can increase by two to four orders of magnitude (from < 200 to ~ 20000 in both dimensions),

and the previous procedure need to be revised so that off-diagonal blocks of these sizes are handled efficiently. The second, longer term, goal of the redesign is to produce a code capable of treating complex atoms in fields with arbitrary polarization. When the field is linearly polarized, rotational symmetry enables a large reduction in complexity and memory usage compared to arbitrary polarizations. With elliptically polarized fields we lose that symmetry, and the RMT data structures in general grow by one to two orders of magnitude in size. In practical terms, we have to integrate the multi-electron Schroedinger equation in higher dimensionality.

To appreciate the scale of the problem, let us consider a typical IR-assisted photoionization problem. In these problems the range of symmetry blocks (angular momenta) treated may be relatively small (0 to 10 for example), but the off-diagonal blocks are typically 2 Gigabytes or greater. Altogether the Hamiltonian in this case would be over 40 gigabytes, and by necessity, would be held in RAM. A typical inner region of the RMT calculation in this case would distribute the problem over thousands of cores. Once the distribution is complete, the Hamiltonian occupies an insignificant amount of RAM per core. The original design of RMT would have read the entire 40 Gigabyte Hamiltonian into a single core, and then distribute parts of it to thousands of inner region cores. In the early versions of RMT, this was the natural approach, since additional numerical processing was required to turn the input data into a Hamiltonian, and the master processor was the only core that had access to additional data files and data structures to perform this processing. Unfortunately, at 40 gigabytes this approach would fail on most machines due to limits on the amount memory available to the master core.

In the case of arbitrary polarization, an additional problem arises. The number of symmetry blocks (angular momenta) increases by one to two orders of magnitude. At present the maximum number of symmetry blocks in the Hamiltonian scales linearly with the maximum angular momentum L : $2 \times (L + 1)$. For elliptical polarization the scaling is quadratic: $2 \times (L + 1) \times (L + 1)$. For $L = 19$, this means an increase from 40 to 800 symmetry blocks. Based on experience with current photoionization calculations, this would require inner region core counts of the order 16,000. More importantly, the Hamiltonian size would increase by two to three orders of magnitude. As a consequence,

the amount of data that would need to be transferred by the master processor would become unmanageable.

We solve both of the problems described above by parallelising the initial input and post-input processing of the Hamiltonian. The off-diagonal blocks of the Hamiltonian matrix are contained in a single multi-gigabyte file 'd', which is created by R-Matrix software [1, 4] written independently of RMT, and which we have not modified as part of this project. Reading in various parts of the 'd' file in parallel by thousands of inner region cores required an excessively complex design, and did not appear to be the most efficient or reliable approach. The problem was therefore solved in several stages. Initially, the data and data structures containing parameters necessary for post-input processing are distributed to all of the inner region cores. The inner region master core then reads the 'd' file incrementally from the file system in sub-gigabyte segments, and writes sub-gigabyte data blocks back to the file system. The inner region cores then read the smaller blocks in parallel, and complete the processing in parallel to create the local Hamiltonians. The first time the program is run it detects the absence of the smaller data blocks, and the inner region master core is told to create them and save to disk. During each subsequent run of the program, the smaller data blocks are detected on disk, a signature file is checked to verify that they were created from the original 'd' file, and then all input and computation is performed in parallel.

This option allows compatibility with the original data file structure. However, modification of the field-free R-matrix codes [1, 4] is now planned as part of general suite development to provide the smaller files directly to RMT if preferred.

The simplest solution also proved to be the most portable. One of the goals was portability between the Cray massively parallel processors, and workstation clusters. To perform the parallel reads mpi_file_read_all, mpi_file_write_all were investigated, but provided no performance improvements. Standard Fortran I/O was judged more likely to be portable. The mpi_file_read_all, for example, behaved unpredictably during tests on the filesystem used in the workstation clusters. Each MPI task is an operating system process, and only one READ per task is attempted, so parallel READs of the files are correctly performed by the operating system. Tests verify that the program runs reliably

and produces identical results on both the Cray, and on the local workstation clusters.

In the class of problems we propose to address, in addition to the off-diagonal blocks of the Hamiltonian matrix, we also had to consider parallel read-in of the channel function information needed to connect the inner and outer regions in the RMT method. This information is contained within the 'Splinewaves' data file. Although it concerns only the diagonal part of the Hamiltonian, it still easily reaches multi-gigabyte sizes in typical problems. The code to read in and distribute the data within this file was redesigned using the same methods described above. The program now reads the file in small increments, distributes the increments to the master processors of each of the symmetry blocks, and reuses the allocated memory. The memory is reclaimed when the process is complete. Only the master cores of the symmetry blocks need this data.

4. Conclusion

To conclude, the communications algorithms of the RMT code have been rewritten in order to reduce memory use, and to work around load balancing problems that occur when the code is applied to interactions between multi-electron atoms and intense linearly polarized light. The developments in the code will be of significant importance for the future development of an RMT code capable of describing multi-electron atoms in intense light fields with arbitrary polarization, including circularly polarized laser light. We have described the completion of phases 1 through 3 as outlined in the workplan associated with the proposal for this eCSE project.

Acknowledgements

This work was funded under the embedded CSE programme of the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk)

References

[1] P.G. Burke. *R-Matrix Theory of Atomic Collisions: Application to Atomic, Molecular and Optical Processes*. Springer Series on Atomic, Optical, and Plasma Physics. Springer Berlin Heidelberg, 2011. ISBN 9783642159312. URL https://books.google.co.uk/books?id=B-jQiI8SQY0C.

[2] L. R. Moore, M. A. Lysaght, J. S. Parker, H. W. van der Hart, and K. T. Taylor. Time delay between photoemission from the $2p$ and $2s$ subshells of neon. *Phys. Rev. A*, 84:061404, Dec 2011. doi: 10.1103/PhysRevA.84.061404. URL http://link.aps.org/doi/10.1103/PhysRevA.84.061404.

[3] L. A. A. Nikolopoulos, J. S. Parker, and K. T. Taylor. Combined r-matrix eigenstate basis set and finite-difference propagation method for the time-dependent schrödinger equation: The one-electron case. *Phys. Rev. A*, 78:063420, Dec 2008. doi: 10.1103/PhysRevA.78.063420. URL http://link.aps.org/doi/10.1103/PhysRevA.78.063420.

[4] C. J. Noble, A. G. Sunderland, and Plummer M. Nag hector dcse report: Combined-multicore parallelism for the uk electron-atom scattering inner region r-matrix codes on hector. http://www.hector.ac.uk/cse/distributedcse/reports/prmat2/, 2012.

[5] Hugo W. van der Hart. B-spline methods for electron scattering and photoionization. http://web.am.qub.ac.uk/ctamop/epi7.php.

[6] Hugo W. van der Hart and Chris H. Greene. Regularities and irregularities in partial photoionization cross sections of he. *Phys. Rev. A*, 66:022710, Aug 2002. doi: 10.1103/PhysRevA.66.022710. URL http://link.aps.org/doi/10.1103/PhysRevA.66.022710.