HECToR optimization of the RMT program

Laura Moore, on behalf of Jonathan Parker

Queen's University Belfast

4th October 2011

Plan of Talk

- Introduction
- The RMT (R-Matrix incorporating Time) method
- Parallelization of RMT
- Scaling studies on HECToR
- Optimization of the outer region
  - Algorithmic enhancement of the time propagator
  - Optimization of workload of first outer region core
- Optimization of the inner region
- Summary

Introduction

Numerical description of laser-atom interactions

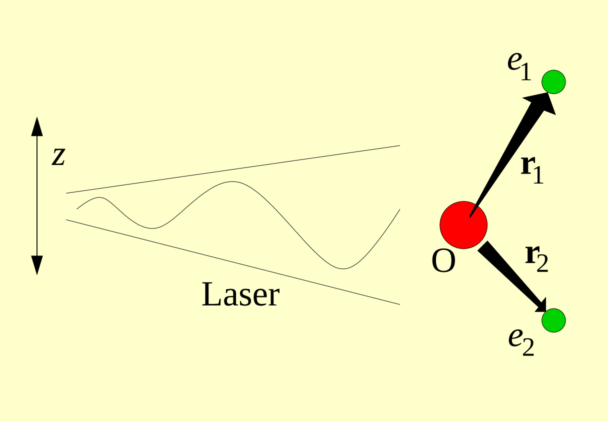
- There is a crucial need to solve the Time-Dependent Schrödinger Equation (TDSE) accurately for multi-electron atoms and molecules coupled to IR/visible/UV and XUV laser fields
- Numerical models need to consider:
  - multi-electron spatial atomic structure
  - multi-electron temporal response to the laser

The Time-Dependent Schrödinger Equation (TDSE)

$$i\hbar\frac{\partial\Psi}{\partial t} = H\Psi$$

- Ψ is the wavefunction describing the electrons
- H is the Hamiltonian describing the interactions that can occur

Helium and a short intense laser pulse



Computational Demand for a 2-electron atom

- 3 coordinates (r, θ, ϕ) for each electron
- Full dimensional treatment for a 2-electron atom achievable on parallel machines
- The HELIUM code: Grown from $\sim$ 16 Gbytes on Cray T3D to $\sim$ 15 Tbytes on Cray XE6

Numerical methods used in the HELIUM code

- (6+1)D PDE – $\Psi(r_1, r_2, \theta_1, \theta_2, \phi_1, \phi_2, t)$
- Finite-difference/ basis set method
- Propagate over grid from $\Psi(\mathbf{r}_1, \mathbf{r}_2, t)$ to $\Psi(\mathbf{r}_1, \mathbf{r}_2, t + \Delta t)$ via an Arnoldi propagator

ES Smyth, JS Parker and KT Taylor 1998 *Comput. Phys. Commun.* **114** 1

Numerical methods used in the HELIUM code

- (6+1)D PDE – $\Psi(r_1, r_2, \theta_1, \theta_2, \phi_1, \phi_2, t)$
- Finite-difference/ basis set method
- Propagate over grid from $\Psi(\mathbf{r}_1, \mathbf{r}_2, t)$ to $\Psi(\mathbf{r}_1, \mathbf{r}_2, t + \Delta t)$ via an Arnoldi propagator

ES Smyth, JS Parker and KT Taylor 1998 *Comput. Phys. Commun.* **114** 1

- The R-matrix concept allows the carry over of HELIUM methods to multi-electron atoms and molecules

The RMT (R-Matrix incorporating Time) method

- A new *ab initio* method to solve accurately the TDSE for multi-electron atoms in intense laser light

Combining HELIUM and R-matrix Methods

- Split into a multi-electron inner region and an outer region in which one electron has become separated from the other electrons

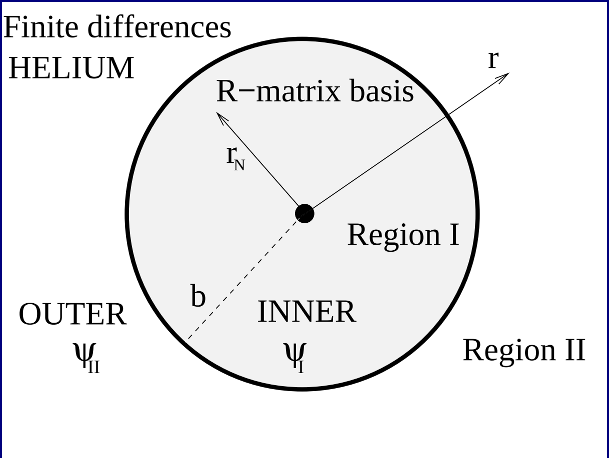
Combining HELIUM and R-matrix Methods

- Split into a multi-electron inner region and an outer region in which one electron has become separated from the other electrons
- Inner region: extension of long-established time-independent R-matrix computer codes to incorporate time dependence

Combining HELIUM and R-matrix Methods

- Split into a multi-electron inner region and an outer region in which one electron has become separated from the other electrons
- Inner region: extension of long-established time-independent R-matrix computer codes to incorporate time dependence
- Outer region: Implementation of HELIUM finite-difference methods

Position Space



Split into an inner region and an outer region

Handling the inter-region boundary

- To solve the TDSE in the outer region: need wavefunction information from the inner region
- and vice-versa!

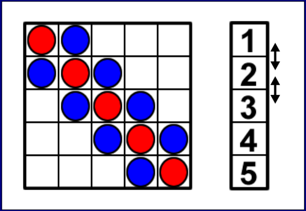
Handling the inter-region boundary

- To solve the TDSE in the outer region: need wavefunction information from the inner region
- and vice-versa!
- At every time-step, cores assigned to the inner region must synchronize with cores assigned to the outer region

Parallelization of RMT

Parallelization of Inner Region

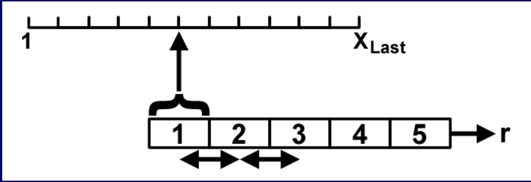
- Calculation involves many Matrix-Vector multiplications where the matrix has block tridiagonal form:



- Parallelize inner region vector over blocks
- Communication limited to neighbouring blocks

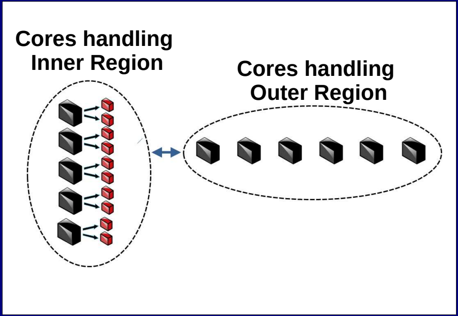
Parallelization of Outer Region

- Calculation involves use of finite difference operators on a grid:



- Parallelize outer region over grid points: each core handles X_{Last} grid points
- Communication limited to neighbouring cores

One-to-one communication between regions



- Different algorithms in the 2 regions
- The 2 regions must synchronize every time-step
- Care with load balancing

Scaling studies on HECToR

Weak Scaling on HECToR

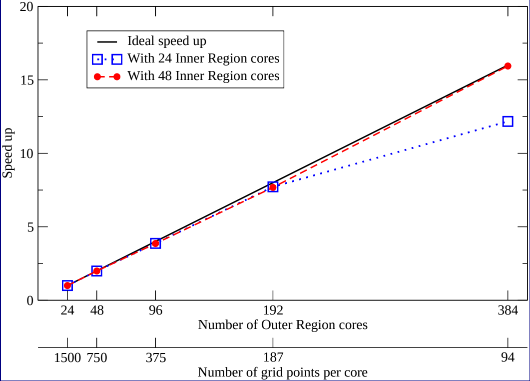
- Number of cores allocated to the Inner Region Group is 24
- X_{Last} is the number of FD points on each Outer Region core

No. cores	No. of FD points	Iter time (s)	
		$X_{Last} = 150$	$X_{Last} = 600$
24	$24 \times X_{Last}$	10.90	43.41
48	$48 \times X_{Last}$	10.98	43.51
96	$96 \times X_{Last}$	10.93	43.66
192	$192 \times X_{Last}$	10.92	43.41
384	$384 \times X_{Last}$	10.93	43.39

- Weak scaling is very good

Strong Scaling on HECToR

- With 94 grid points per Outer Region core, load balancing improves when the Inner Region is allocated 48 cores



Optimization of the outer region

Optimization of the outer region

1. Algorithmic enhancement of time propagator

Scaling of the time-step δt

- Highest energy eigenvalues of the FD Hamiltonian are of the order $E = \hbar^2 k^2 / 2m$ where $k^2_{\mathrm{max}} = \pi^2 / \delta r^2$
- The propagator must integrate the equations of motion as though these high-energy unphysical modes contain population
- In the explicit propagator used by RMT, outer region stability requires time-steps δt that scale as $1/\delta r^2$

The eigenspectrum of the outer region Hamiltonian

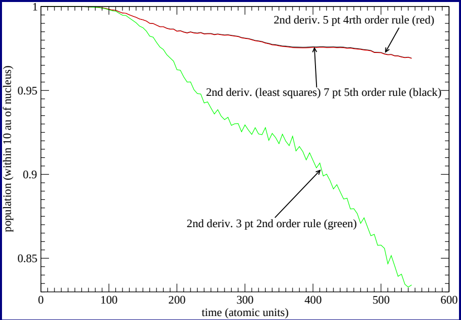
- Dominated by the kinetic energy operator K, which in turn is dominated by the 2nd derivative operator
- Expect that δt is governed by $1/\delta r^2$ dependence of 2nd derivative operator
- But both inner and outer regions use same δt
  - Behaviour of eigenspectrum of K has profound effect on run-time efficiency of RMT

Goal:

- Develop methods to mitigate the effect of high eigenvalues of K
  - New technique to reduce the peak eigenvalues of K by using least squares operators
  - Successfully reduces peak eigenvalues by up to a factor of 4 with very little additional computational cost

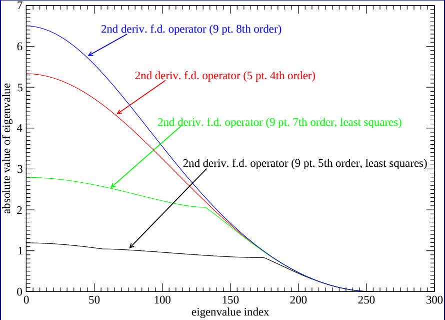
Integration using 3 different 2nd derivative operators

- Failure of standard 3 point rule for 2nd order derivatives
- Success of a least-squares finite-difference rule (7 point)



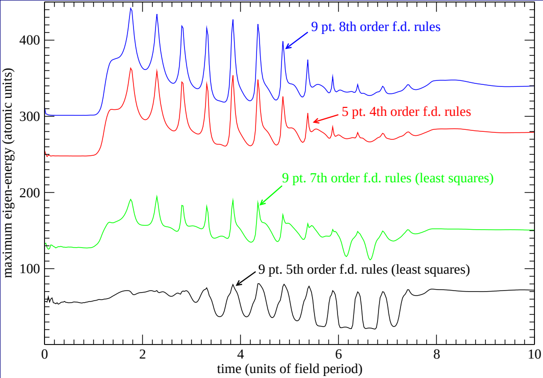
Eigenspectra of 4 FD 2nd derivative operators

- Least squares process can dramatically truncate the higher frequency components of the eigenspectrum without significantly changing the low frequency components



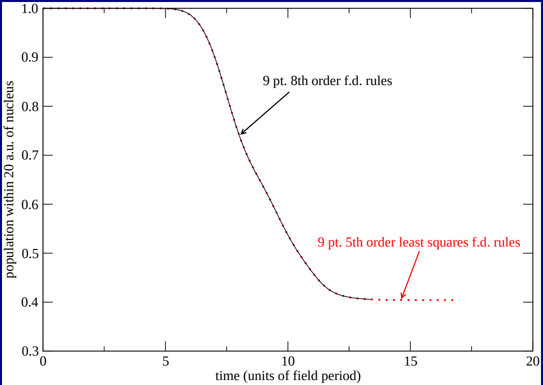
Maximum eigenenergy of outer region Hamiltonian during the course of an 8 field period integration

- Least squares operators have the smallest maximum eigenenergies



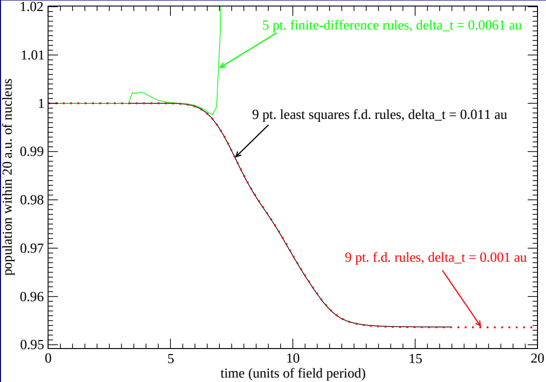
Population within 20 Bohr radii of the nucleus as a function of time

- Least squares rules give same accuracy as non least-squares rules



Increase in δt achievable using least squares rules

- At $\delta t = 0.0061$ a.u. 5 point rule fails
- The 9 point least squares rule is correct at $\delta t = 0.011$ a.u.



Run time efficiency of 9 point least squares rules

- On HECToR, 9 point rules nearly as fast as 5 point rules - overhead is in accessing arrays from memory
- Memory fetch overhead is the same for 9 and 5 point rules
- 2nd derivative operator has small computational cost in comparison to other operators in the Hamiltonian
- Ability to increase δt by a factor 1.8 directly translates into a factor 1.8 increase in integration speed

Optimization of the outer region

Optimization of the outer region

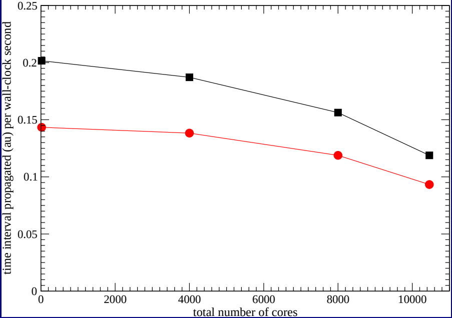
2. Optimization of workload of first outer region core

Optimization of workload of first outer region core

- First outer region core has additional workload compared to other outer region cores (calculation of boundary terms to be sent to the inner region)
- Adapt RMT so that the first outer region core handles a reduced number of grid points
- With the optimum workload for the first outer region core, it synchronizes with the remaining outer region cores every time-step

Time interval propagated per wall-clock second

- The first outer region core handles the same number (RED) and a reduced number (BLACK) of grid points compared to other outer region cores



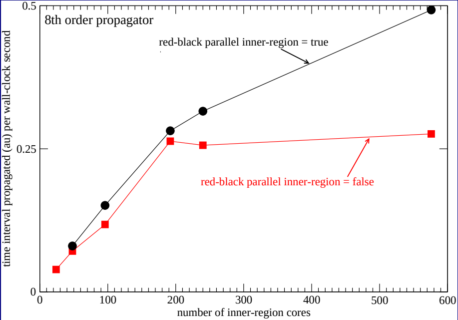
Optimization of the inner region

Inner region cores to handle even and odd orders of propagation

- The inner and outer regions synchronize at the start and end of every time-step
- Arnoldi propagators are used in both regions
- The maximum propagation order is set as a parameter
- At every order of propagation within a time-step, the inner region requires starting information sent from the outer region
- Divide inner region into 2 independent sets of cores - red and black - which independently receive information from the outer region

Time interval propagated per wall-clock second

- Near linear speed up to about 240 inner region cores
- Above 240 cores improvement in speed continues linearly (but more slowly)



Inner region cores to handle even and odd orders of propagation

- Addition of 340 inner region cores (above the 240 core threshold) gives 70% improvement in speed
- Both desirable and beneficial in RMT - outer region runs on 1000's or tens of 1000's of cores
- Addition of 340 inner region cores is a highly inexpensive way to improve the run-time efficiency of the entire program by 70%

Summary

Summary

- ✓ Enhanced time propagator successfully implemented
  - using least squares processes in 2nd derivative operators has the benefit of lowering peak eigenenergies, thus allowing a larger time-step which can increase integration speed by a factor of 80%

Summary

- ✓ Enhanced time propagator successfully implemented
  - using least squares processes in 2nd derivative operators has the benefit of lowering peak eigenenergies, thus allowing a larger time-step which can increase integration speed by a factor of 80%
- ✓ Optimum workload for first outer region core successfully implemented
  - reducing the number of grid points handled by the first outer region core can speed up the calculation by around 20%

Summary

- ✓ Enhanced time propagator successfully implemented
  - using least squares processes in 2nd derivative operators has the benefit of lowering peak eigenenergies, thus allowing a larger time-step which can increase integration speed by a factor of 80%
- ✓ Optimum workload for first outer region core successfully implemented
  - reducing the number of grid points handled by the first outer region core can speed up the calculation by around 20%
- ✓ Even/odd propagation orders handled by independent sets of inner region cores successfully implemented
  - adding a few 100 inner region cores can speed up, by 70% or more, entire calculations using 1000's of cores in total

Thanks for your attention!