Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 221 (2007) 148–157
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Solution of time-independent Schrödinger equation by the imaginary time propagation method

L. Lehtovaara [a], J. Toivanen [b], J. Eloranta [a,c,*]

[a] Department of Chemistry, University of Jyväskylä, Survontie 9, 40500 Jyväskylä, Finland
[b] Center for Research in Scientific Computation, Box 8205, North Carolina State University, Raleigh, NC 27695-8205, USA
[c] Department of Chemistry and Biochemistry, California State University Northridge, Northridge, CA 91330-8262, USA

Received 11 April 2006; received in revised form 5 June 2006; accepted 7 June 2006
Available online 25 July 2006

Abstract

Numerical solution of eigenvalues and eigenvectors of large matrices originating from discretization of linear and non-linear Schrödinger equations using the imaginary time propagation (ITP) method is described. Convergence properties and accuracy of 2nd and 4th order operator-splitting methods for the ITP method are studied using numerical examples. The natural convergence of the method is further accelerated with a new dynamic time step adjustment method. The results show that the ITP method has better scaling with respect to matrix size as compared to the implicitly restarted Lanczos method. An efficient parallel implementation of the ITP method for shared memory computers is also demonstrated.

Keywords: Imaginary time propagation; Schrödinger equation; Lanczos method; Operator-splitting

1. Introduction

Our recent efforts in solving the "electron in helium" – problem numerically indicated the need for a systematic study of fast methods for diagonalizing matrices ("Hamiltonians") that originate from three-dimensional linear and non-linear Schrödinger equations [1,2]. With this particular problem we noticed that, for example, a Gaussian-type basis set was not sufficiently flexible for producing accurate results and hence it was necessary to solve the problem using a less restrictive grid basis [2]. In 3D the size of the Hamiltonian matrix grows rapidly as a function of the number of spatial grid points and the problem size quickly increases beyond computational resources. In the present case, with spatial grid sizes extending up to $512 \times 512 \times 512$ only iterative methods, such as variants of the Lanczos method and the imaginary time propagation of Schrödinger equation (i.e. by mapping time $t \to -it$; ITP), are applicable in practice. In this paper we will carry out comparison of the implicitly restarted Lanczos method, as implemented in the ARPACK package, and

* Corresponding author. Address: Department of Chemistry, University of Jyväskylä, Survontie 9, 40500 Jyväskylä, Finland. Tel.: +358 50 373 2122; fax: +358 14 260 2551.
E-mail address: Jussi.Eloranta@csun.edu (J. Eloranta).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.06.006

the ITP method with 2nd and 4th order time propagators. The main aim is to find a practically applicable method for solving large-scale matrix eigenproblems that result from discretization of linear and non-linear Schrödinger equations [3].

Both the implicitly restarted Lanczos and ITP methods have been extensively studied previously (see, for example, Refs. [3–7] and references therein). The Lanczos method is used as a reference of comparison in the current work and hence the basic details of this method are not given here. Instead, we concentrate on formulation and discussion of the ITP method. This method is, for example, a key element in many Monte Carlo-based approaches. Unlike with Monte Carlo, it is possible to obtain excited states when the resulting diffusion-type problem is solved by non-stochastic methods. This can be carried out by propagating a number of different wave functions and orthogonalizing them after each time step [5]. The ITP method can be efficiently implemented in 3D by using the operator splitting technique. The splitting of the kinetic and potential parts in Schrödinger equation involves an error that is proportional to the power of the time step and depends on the accuracy of the splitting scheme. Recently, Krotscheck et al. studied convergence properties of split operators with various orders [4,5]. With a given splitting scheme, the exponential operators must be propagated over short time steps. This can be carried out conveniently in coordinate and momentum spaces for the potential and kinetic operators, respectively. Transformation between the spaces can be computed efficiently by the fast Fourier transform (FFT)-based methods. It should be recalled that the use of FFT implies periodic boundary conditions. The ITP method is also well suited for solving non-linear Schrödinger equations whereas, for example, the Lanczos based methods typically employ a self-consistent-field (SCF; "fixed point iterations")-type approach.

In this paper, we (I) compare convergence properties and accuracy of the 2nd and 4th order operator-splitting methods for ITP, (II) study the effect of dynamic time step adjust for convergence, (III) obtain practical computational scaling for the applied ITP method, (IV) compare the ITP and implicitly restarted Lanczos methods, and (V) demonstrate that an efficient parallel version of the ITP algorithm can be implemented. Both linear and non-linear problems are considered in this work.

2. The imaginary time propagation method

We are seeking approximate solutions to time-independent Schrödinger equation:

$$\begin{aligned} &H\psi_i(r) = E_i\psi_i(r) \quad i = 1, 2, 3, \ldots \\ &H = T + V(x) = -\frac{\hbar^2}{2m}\Delta + V(x) \end{aligned} \tag{1}$$

where $\hbar$ is the Planck's constant (one in atomic units), H is the Hamiltonian consisting of kinetic energy operator T and potential energy operator V, ψ_i are the eigenfunctions, and E_i the eigenvalues. The potential energy operator V can be either linear or non-linear. If the eigenfunctions ψ_i are known, the eigenvalues can be calculated from the expectation value $E_i = \langle\psi_i|H|\psi_i\rangle$. This means that it is possible to modify the original problem in such a way that the eigenfunctions remain unchanged but the eigenvalues are altered. One such approach is to use the inverse iteration method, where the original Hamiltonian is replaced by $(H + \alpha)^{-1}$. The shift α is chosen such that the operator is positive definite and has good spectral properties [5,8]. Another approach is based on using the corresponding time-dependent Schrödinger equation in imaginary time ($t = -\mathrm{i}\tau$):

$$\frac{\partial\psi(r,\tau)}{\partial\tau} = -\frac{H}{\hbar}\psi(r,\tau) \tag{2}$$

where $\psi(r,\tau)$ is a wavefunction that is given by a random initial guess at $\tau = 0$ and converges towards the ground state solution $\psi_0(r)$ when $\tau \to \infty$. This is the key equation in the ITP method. The formal solution to this equation can be written as ($\tau = n\Delta\tau$):

$$\psi(r,\tau) = \mathrm{e}^{-H\tau/\hbar}\psi(r,0) = (\mathrm{e}^{-H\Delta\tau/\hbar})^n\psi(r,0) \tag{3}$$

In fact, the above exponential operator $\exp(-H\tau/\hbar)$ has the same eigenfunctions as Eq. (1). The last form in Eq. (3) can be thought as an analog of the power method or its generalization the subspace iteration, which can also be applied for the inverse iteration Hamiltonian [9]. Krotscheck et al. have carried out a compar-

ison of both modified Hamiltonians using the power iteration scheme and observed that the exponential form has much faster convergence [5]. The inverse iteration Hamiltonian can be considered as the lowest level approximation to the exponent operator Taylor expansion. In order to carry out practical calculations with the exponential-form operator, it is necessary to decouple the kinetic and potential parts from each other using the operator splitting technique [5,7,10]. Unfortunately, for most problems the splitting cannot be done exactly and one has to resort to approximation. In this work we will consider two different schemes [5,11,12]:

$$\begin{aligned} \mathrm{e}^{-(T+V)\Delta\tau/\hbar} &\approx \mathrm{e}^{-\frac{1}{2}V\Delta\tau/\hbar} \times \mathrm{e}^{-T\Delta\tau/\hbar} \times \mathrm{e}^{-\frac{1}{2}V\Delta\tau/\hbar} + \mathrm{O}(\Delta\tau^3) \quad \text{(2nd order)} \\ \mathrm{e}^{-(T+V)\Delta\tau/\hbar} &\approx \mathrm{e}^{-\frac{1}{6}V\Delta\tau/\hbar} \times \mathrm{e}^{-\frac{1}{2}T\Delta\tau/\hbar} \times \mathrm{e}^{-\frac{2}{3}\left(V+\frac{1}{48}[V,[T,V]]\Delta\tau^2\right)\Delta\tau/\hbar} \times \mathrm{e}^{-\frac{1}{2}T\Delta\tau/\hbar} \times \mathrm{e}^{-\frac{1}{6}V\Delta\tau/\hbar} + \mathrm{O}(\Delta\tau^5) \quad \text{(4th order)} \end{aligned} \tag{4}$$

The commutator in the 4th order expression can be written as:

$$[V,[T,V]] = \frac{\hbar^2}{m}|\nabla V|^2 \tag{5}$$

This term can be evaluated either by calculating the gradient accurately in the momentum space or by using the finite difference approximation. Transformation between the coordinate and momentum spaces can be carried out by using the fast Fourier transformation (FFT) technique [13]. Propagation of each split component in Eq. (4) can be efficiently evaluated in the space where the component operator is diagonal (i.e. kinetic operator T in the momentum and potential operator V in coordinate space). In the view of power iteration, the split operator technique allows for efficient calculation of the matrix exponential in Eq. (3).

The basic ITP method yields only the ground state solution to Eq. (1). The excited states can be obtained by propagating a number of different wavefunctions simultaneously in time and requiring that the states are orthonormal after each time step. This can be achieved, for example, by Gram–Schmidt orthogonalization or by diagonalizing the overlap matrix. For a small number of states both methods were accurate, however, when the number of states increased the Gram–Schmidt method became numerically inaccurate. Thus in this work we orthogonalize the states by diagonalizing the overlap matrix:

$$\begin{aligned} &SC = \lambda C \\ &S_{pq} = \langle \Psi_p(\tau+\Delta\tau)|\Psi_q(\tau+\Delta\tau)\rangle = \int \Psi_p^*(r)\Psi_q(r)d^3r = \Delta^3\sum_i \Psi_p^*(r_i)\Psi_q(r_i) \\ &\Psi_q(\tau+\Delta\tau) = \mathrm{e}^{-H\Delta\tau/\hbar}\phi_q(\tau) \end{aligned} \tag{6}$$

where S is the overlap matrix, C is a matrix of the orthogonalized states, ϕ_q are the orthonormal wave functions from the previous diagonalization, λ are the corresponding eigenvalues, Δ is the spatial grid step in the discrete problem and r_i's are the corresponding grid points.

Estimates for the eigenvalues can be calculated during the ITP from the corresponding expectation values or from the time decay of the wavefunctions [14]:

$$\begin{aligned} E_H^i(\tau+\Delta\tau) &= \langle\phi_i(\tau+\Delta\tau)|H|\phi_i(\tau+\Delta\tau)\rangle = E_i + \mathrm{O}(\Delta\tau^{2\alpha}) \\ E_D^i(\tau+\Delta\tau) &= -\frac{\ln(\langle\Psi_i(\tau+\Delta\tau)|\Psi_i(\tau+\Delta\tau)\rangle)}{2\Delta\tau/\hbar} = E_i + \mathrm{O}(\Delta\tau^{\alpha}) \end{aligned} \tag{7}$$

where E_i is the closest eigenvalue, α is the order of error in the wave function, τ is the current time position, and $\Delta\tau$ is the time step. An estimate for the relative error can be calculated as follows:

$$\Delta\varepsilon_i = \frac{\Delta E_i}{E_i} \approx \frac{\Delta E_i}{E_H^i} \tag{8}$$

when E_H is close to E_i and ΔE_i is the absolute error. From the eigenvalue error bound in Ref. [15], it follows that standard deviation of the Hamiltonian can be used in estimating the absolute error:

$$\Delta E_i = |E_i - \langle\psi_i|H|\psi_i\rangle| \leqslant \sqrt{2}\sqrt{\langle\psi_i|H^2|\psi_i\rangle - \langle\psi_i|H|\psi_i\rangle^2} \tag{9}$$

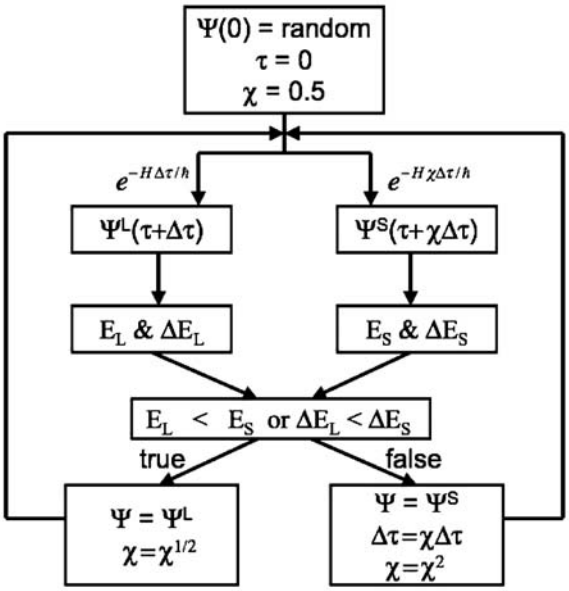

Fig. 1. A flow chart of the ITP algorithm involving the dynamic time step adjustment. When more than one state is included, E_L, E_S, ΔE_L and ΔE_S were calculated as a root-mean-square of the corresponding values for individual eigenvalues.

where the trial wavefunction ψ_i is close to eigenstate i. This error estimate can be used as a criterion for adjusting the imaginary time step during the time propagation. However, Eq. (9) usually leads to too tight error bound. For this reason the absolute error of the ITP method was based on the difference between the accurate and approximate eigenvalues. The convergence speed of the power iteration method is proportional to $|E_k/E_{k+1}|^\tau$ [15]. When τ and $|E_k - E_{k+1}|$ are large, we have for a given state k ($|C_k| \approx 1$, $|C_{k+1}| \ll 1$ and all other coefficients are close to zero):

$$\begin{aligned}
&\langle\psi_k|H^2|\psi_k\rangle - \langle\psi_k|H|\psi_k\rangle^2 = \frac{1}{2}\sum_{i\neq j} C_i^2 C_j^2 (E_i - E_j)^2 \approx C_{k+1}^2 |E_k - E_{k+1}|^2 \\
&|E_k - \langle\psi_k|H|\psi_k\rangle| \approx C_{k+1}^2 |E_k - E_{k+1}| \\
&\Rightarrow \frac{\langle\psi_k|H^2|\psi_k\rangle - \langle\psi_k|H|\psi_k\rangle^2}{|E_k - E_{k+1}|} \approx |E_k - \langle\psi_k|H|\psi_k\rangle|
\end{aligned} \tag{10}$$

where ψ_i are approximate eigenvectors from ITP, coefficients C are expansion coefficients of ψ_i in the eigenvectors of H, E_i are the exact eigenvalues of H. When the energy difference $|E_k - E_{k+1}|$ is known to be sufficiently large, the right-hand side in Eq. (9) can be squared to obtain a convergence estimate for ITP.

The basic scheme for time step adjustment algorithm is shown in Fig. 1. The main idea is to propagate using two different time steps and reduce the step only if this would produce a smaller absolute error and a smaller eigenvalue. The current algorithm is only allowed to make the time step shorter and hence it is important to choose the initial time step sufficiently large. However, if too large initial time step is chosen, the ITP process becomes numerically unstable.

3. Numerical results and discussion

All serial calculations were carried out on PowerPC G5 based computers. Parallelization of the time integration scheme was implemented using OpenMP directives [16] and the resulting code was executed on IBM pSeriers 690 shared memory computer using up to 11 processors.

3.1. 3D linear Schrödinger equation with an anharmonic potential

The following anharmonic form was used as a linear potential in Eq. (1):

$$V_{\mathrm{lin}}(x,y,z) = \Delta\left(1 - \cos^2\left(\frac{\pi x}{k_x}\right)\cos^2\left(\frac{\pi y}{k_y}\right)\cos^2\left(\frac{\pi z}{k_z}\right)\right) \tag{11}$$

where $\Delta = 10$ Hartree and $k_x = k_y = k_z = 6.4$ Bohr. It should be noted that the use of purely harmonic potential might lead to an accidental cancellation of terms and therefore it is not a good choice for test purposes. Unless otherwise noted the standard grid size was $32 \times 32 \times 32$, the spatial grid step 0.2 Bohr and the initial time step 1. Atomic units are used through out.

In the first set of runs accuracy of the 2nd and 4th order methods is demonstrated (see Fig. 2). The 2nd order method (see Eq. (4)) reaches 4th order in energy and the 4th order correspondingly 8th order in energy. In the latter case floating point precision was an important factor. This can be seen by comparing the single and double precision results in Fig. 2. For smaller time steps than shown in the figure, double precision also runs out of accuracy and a sudden drop in the curve occurs. The position of the drop depends on the form of the potential operator. Convergence of the four lowest states is shown in Fig. 3. Discontinuities in the curves result from the dynamic time step adjust procedure. The difference between the methods becomes significant when accurate results are required. For example, to reach relative errors less than 10^{-5}, the 2nd order method requires four times more iterations than the 4th order method. Iterations using the 4th order method are approximately twice as expensive to compute, but due to smaller number of iterations required, it consumes less CPU time overall in reaching the desired accuracy.

When degenerate eigenstates are present, it is important to include all such states in the calculation even if just one of them is sought. This is demonstrated in Fig. 4 where two identical calculations were carried out with four and five states included. The fifth state belongs to a degenerate set of states not fully included in the calculation. In both cases the first four states converge rapidly but the fifth state converges extremely slowly. If the rest of the states from this particular degenerate set were included in the calculation, fast convergence of this state would be restored. However, it should be noted that inclusion of more states in the calculation, increases the computational cost of the ITP algorithm. This is shown in Fig. 5 where wall clock time was obtained as a function of number of states included. The scaling is $O(N^2)$, where N is the number of states. This scaling results form the underlying orthogonalization procedure.

Finally, comparison of the ITP and the implicitly restarted Lanczos (ARPACK) method was carried out. In the latter case the kinetic energy operator T was also evaluated using FFT and the number of Arnoldi vectors was chosen to be twice the number of eigenstates. Wall clock time as a function of grid points is shown in

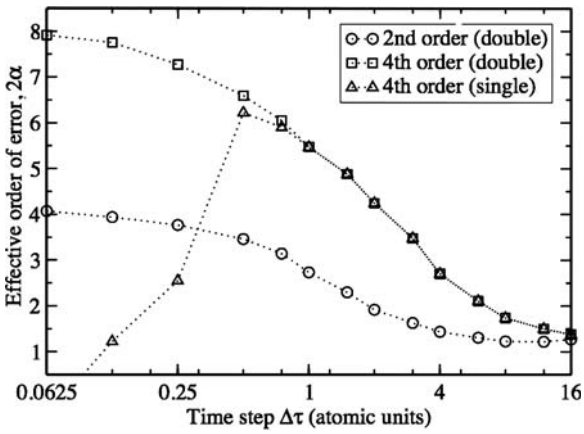

Fig. 2. Accuracy of the 2nd and 4th order operator splitting formulas (Eq. (4)). The matrix problem originates from discretization of the linear Schrödinger equation as described in Section 3.1. The 2nd order formula (circles) reaches 4th order accuracy in energy whereas the 4th order formula (squares and triangles) reaches 8th order in energy as the time step is reduced. Note the difference between the single (triangles) and double precision (squares) results and that the x-axis is logarithmic. For definition of 2α, see the first line of Eq. (7).

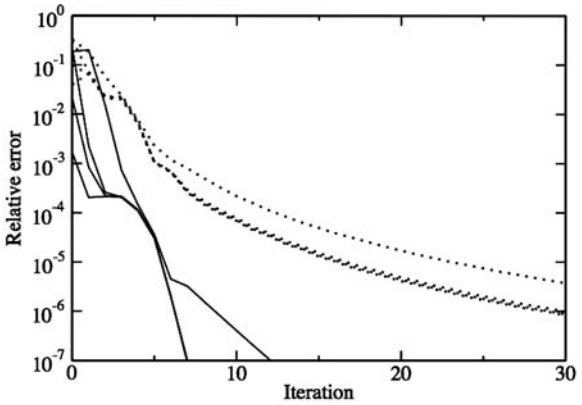

Fig. 3. Convergence of the 2nd and 4th order formulas (Eq. (4)) as a function of number of time steps. The matrix problem originates from discretization of the linear Schrödinger equation as described in Section 3.1. The solid lines represents the convergence of the four lowest energy states using the 4th order formula and the dotted lines the 2nd order formula. Note that the y-axis is logarithmic.

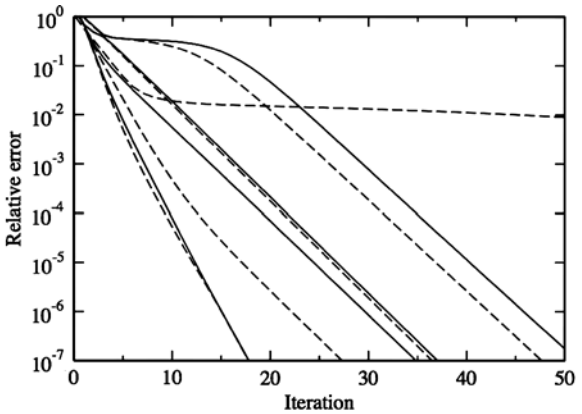

Fig. 4. The effect of degeneracies on the convergence of the ITP method. The matrix problem originates from discretization of the linear Schrödinger equation as described in Section 3.1. The solid lines corresponds to the four lowest states with all degenerate states included in the calculation and dashed lines to the five lowest states with only one state from the next degenerate set of states. Note the poor convergence of the fifth state and that the y-axis is logarithmic.

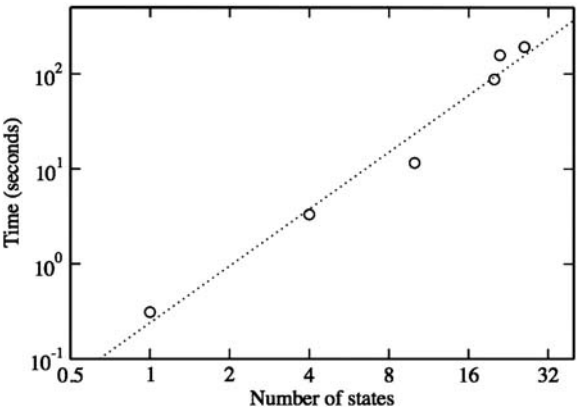

Fig. 5. Scaling of the ITP method as a function of states included in the calculation. The matrix problem originates from discretization of the linear Schrödinger equation as described in Section 3.1. Circles mark the computed points and the dotted line denotes a linear fit to the logarithmic data. Note that both axes are logarithmic.

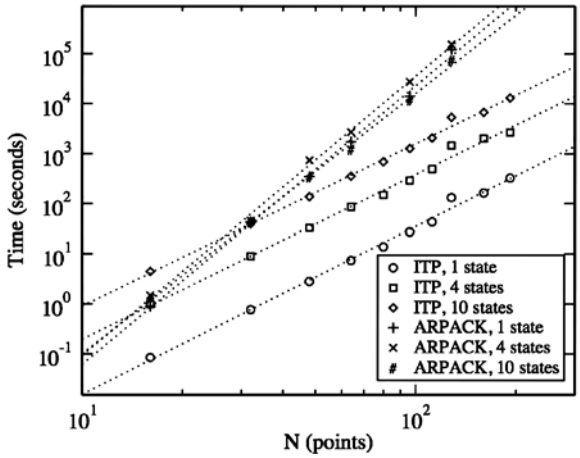

Fig. 6. Computational scaling of the ITP and implicitly restarted Lanczos methods (ARPACK) for 1, 4 and 10 states. The matrix problem originates from discretization of the linear Schrödinger equation as described in Section 3.1. The Lanczos method results in a slope of $k \sim 5.5$ and the ITP method $k \sim 3.3$. The dotted lines are linear fits to the data. The matrix dimension is $N^3 \times N^3$. Note that both axes are logarithmic.

Fig. 6 with 1, 4 and 10 states included in the calculation. The end criterion for the ITP method was set to $\Delta E < 10^{-7}$ whereas machine precision had to be used for ARPACK. In the latter case the choice was based on the observation that ARPACK would otherwise miss eigenstates. The ITP method always found the lowest states properly with the given accuracy. The slopes for the Lanczos and ITP methods were $k \sim 5.5$ and $k \sim 3.3$, respectively, as obtained from the dotted lines in Fig. 6. This demonstrates clearly the ITP method has much better scalability in terms of the grid size than the Lanczos method. On the other hand, the Lanczos method scales much better as function of number of states. For large problems the overall scaling makes the ITP method faster when a small number of eigenstates are requested. A parallel implementation of the ITP method for a shared memory machine was implemented using the standard OpenMP tools [16]. Parallelization of the main for – loops yields already good results as shown in Fig. 7. The effective speed-up with 11 processors was close to 9, which yields 82% total efficiency.

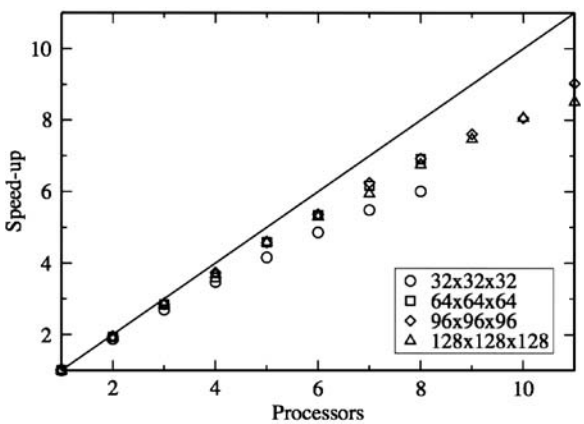

Fig. 7. Parallel scaling of the ITP method. The matrix problem originates from discretization of the linear Schrödinger equation as described in Section 3.1. The y-axis indicates how many times faster (“speed-up”) the code ran as compared to the single processor case. The solid line shows the theoretical maximum.

3.2. 3D non-linear Schrödinger equation with a Gross–Pitaevskii potential

The following non-linear potential was used in Eq. (1):

$$V_{\text{non-lin}}(\psi; x, y, z) = \lambda|\psi(x, y, z)|^2 + V_{\text{lin}}(x, y, z) \tag{12}$$

where $\lambda = 6.86 \times 10^{-3}$ Hartree Bohr^{-3} (i.e. 7 K/0.0218 Å^{-3}) and V_{lin} is given by Eq. (11) with $k_x = k_y = k_z = 25.6$ Bohr and $\Delta = 3.17 \times 10^{-4}$ Hartree (i.e. 100 K). The non-linear part resembles the Gross–Pitaevskii potential for superfluid ^{4}He and the linear part corresponds to an anharmonic trap (see Section 3.1). Unless otherwise noted the standard grid size was $32 \times 32 \times 32$ and the spatial grid step 0.8 Bohr.

Convergence of the relative error (Eq. (9)) using two different initial time steps is shown in Fig. 8. The initial time step of 10^4 was sufficiently large to achieve rapid initial convergence whereas the smaller time step resulted in clearly slower initial convergence. In practice the initial time step should be chosen as large as possible. However, too large time step result in unstable time propagation. A similar problem appears with ARPACK when too aggressive updating of the current SCF solution is used. Behavior of the dynamic time step adjustment procedure is demonstrated in Fig. 9. It is clearly seen that the small initial time step causes

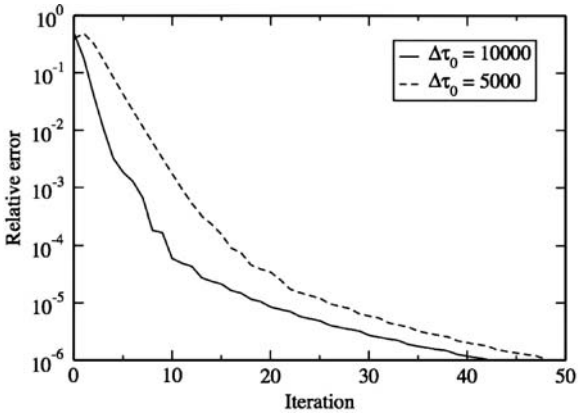

Fig. 8. The effect of dynamic time step adjust on convergence (relative error) of the ITP method. The matrix problem originates from discretization of the non-linear Schrödinger equation as described in Section 3.2. The solid line indicates the initial time step of 10^4 and the dashed line 5×10^3. Discontinuities in the curves arise from the time step adjustment. Note that the y-axis is logarithmic.

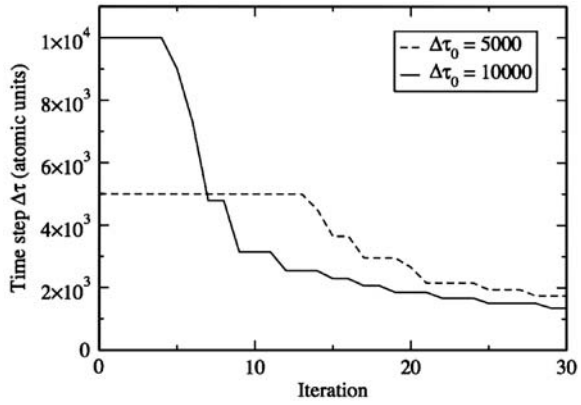

Fig. 9. Dynamic time step adjust during the ITP iterations. The matrix problem originates from discretization of the non-linear Schrödinger equation as described in Section 3.2. The solid line indicates initial time step of 10^4 and the dashed line 5×10^3. Jumps in the curves arise from time step adjustments.

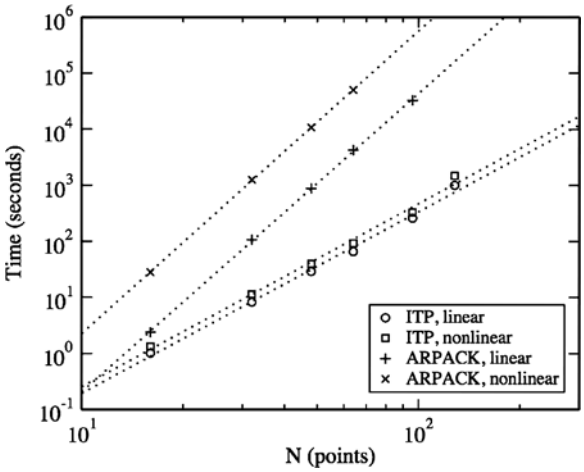

Fig. 10. Computational scaling of the ITP and implicitly restarted Lanczos methods (ARPACK) for linear and non-linear problems. The matrix problem originates from discretization of the non-linear Schrödinger equation as described in Section 3.2. The dotted lines are linear fits to the data. The matrix dimension is $N^3 \times N^3$. Note that both axes are logarithmic.

excessive propagation with non-optimal time step. If a good diagnostics for problem stability would exist, it would also be possible to increase the time step in the adjustment algorithm.

For non-linear problems an SCF-type approach must be used with ARPACK since it does not allow for changing the underlying matrix during the iterations. Thus sequences of linear problems are solved which then converge towards the correct non-linear problem as well as its solution. In the ITP method the matrix can change during the imaginary time evolution and therefore the non-linearity can be treated in a natural way. Results from comparing implicitly restarted Lanczos and the ITP methods are shown in Fig. 10. Calculations were carried out using 10^{-7} relative error convergence limit (i.e. exact form of Eq. (9)). Furthermore, in the ARPACK case, the SCF iterations were terminated when the relative difference between the successive eigenvalue estimates fell below the 10^{-7} limit. This standard termination condition does not guarantee that the error would yet be below the required bound. Computational scaling with respect to the number of grid points remains unchanged going from a linear (Section 3.1) to a non-linear problem. Small increase in the prefactor was observed for the ITP method and a larger increase for the Lanczos method (see Fig. 10). In the latter case the increase originates from the explicit SCF treatment. For large problems, however, these differences are insignificant. In general, the ITP method preserves the better computational scaling as compared to Lanczos.

4. Conclusions

We have shown that the ITP method is a fast method for obtaining eigenvectors and eigenvalues of large matrices that originate from discretization of linear and non-linear Schrödinger equations. Best convergence properties were achieved by using the 4th order propagator with the dynamic time step adjustment. It is important that all states from possible degenerate sets are included in the calculation in order to obtain fast convergence. Our numerical tests show that the ITP method has much better scaling as a function of matrix size as compared to the implicitly restarted Lanczos method (ARPACK). In fact, solution of our "electron in helium" problem using a large grid basis ($512 \times 512 \times 512$) was practically impossible using the Lanczos method due to the required long computational time. The ITP method can also treat the non-linearity in this problem naturally without any SCF treatment. Parallel implementation of the ITP method using a shared memory computer and a small number of processors is straightforward. Scaling to higher number of processors would presumably just require more detailed parallelization of the code. For our current problem scaling up to 11 processors is sufficient.

Acknowledgments

Research funding by University of Jyväskylä and the Finnish Academy of Sciences, and computational resources by Center for Scientific Computing (CSC) are acknowledged.

References

[1] J. Eloranta, V.A. Apkarian, A time dependent density functional treatment of superfluid dynamics: equilibration of the electron bubble in superfluid ^{4}He, J. Chem. Phys. 117 (2002) 10139.
[2] L. Lehtovaara, J. Eloranta, Small multielectron bubbles in bulk superfluid ^{4}He, in: Proceedings of 24th International Conference on Low Temperature Physics, in press.
[3] R.B. Lehoucq, D.C. Sorensen, P.A. Vu, C. Wang, ARPACK: an implementation of an implicitly restarted Arnoldi method for computing some of the eigenvalues and eigenvectors of a large sparse matrix, Dept. Comput. Appl. Math., Rice University, 1996. For further details see ⟨http://www.caam.rice.edu/software/ARPACK⟩.
[4] J. Auer, E. Krotscheck, S.A. Chin, A fourth-order real-space algorithm for solving local Schrödinger equations, J. Chem. Phys. 115 (2001) 6841.
[5] M. Aichinger, E. Krotscheck, A fast configuration space method for solving local Kohn–Sham equations, Comput. Mater. Sci. 34 (2005) 188.
[6] A.K. Roy, N. Gupta, D.M. Deb, Time-dependent quantum-mechanical calculation of ground and excited states of anharmonic and double-well oscillators, Phys. Rev. A 65 (2001) 012109.
[7] L. Lehtovaara, T. Kiljunen, J. Eloranta, Efficient numerical methods for simulating static and dynamic properties of superfluid helium, J. Comp. Phys. 194 (2004) 78.
[8] J. Martikainen, T. Rossi, J. Toivanen, Computation of a few smallest eigenvalues of elliptic operators using fast elliptic solves, Commun. Numer. Meth. Eng. 17 (2001) 521.
[9] E. Isaacson, H.B. Keller, Analysis of Numerical Methods, Dover Publications, Inc., New York, 1994.
[10] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, second ed., Cambridge University Press, Cambridge, UK, 1999.
[11] M. Suzuki, in: D.P. Landau, K.K. Mon, H.-B. Shüttler (Eds.), Computer Simulation Studies in Condensed Matter Physics, vol. VIII, Springer, Berlin, 1996.
[12] S.A. Chin, Phys. Lett. A 226 (1997) 344.
[13] M. Frigo, S.G. Johnson, FFTW: an adaptive software architecture for the FFT, ICASSP Conference Proceedings 3 (1998) 1381, For more details see http://www.fftw.org.
[14] L. Lehtovaara, J. Eloranta, A 2-level anisotropic electronic system in superfluid ^{4}He, J. Low Temp. Phys. 138 (2005) 91.
[15] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., John Hopkins University Press, Baltimore, MD, 1996.
[16] OpenMP C and C++ application program interface. For further details see http://www.openmp.org.