New, Highly Accurate Propagator for the Linear and Nonlinear Schrödinger Equation

Hillel Tal-Ezer, Ronnie Kosloff & Ido Schaefer

Journal of Scientific Computing

ISSN 0885-7474

J Sci Comput
DOI 10.1007/s10915-012-9583-x

Springer

J Sci Comput
DOI 10.1007/s10915-012-9583-x

New, Highly Accurate Propagator for the Linear and Nonlinear Schrödinger Equation

Hillel Tal-Ezer · Ronnie Kosloff · Ido Schaefer

Received: 17 August 2011 / Revised: 6 January 2012 / Accepted: 19 February 2012

Abstract A propagation method for the time dependent Schrödinger equation was studied leading to a general scheme of solving ode type equations. Standard space discretization of time-dependent pde's usually results in system of ode's of the form

$$u_t - Gu = s \tag{0.1}$$

where G is a operator (matrix) and u is a time-dependent solution vector. Highly accurate methods, based on polynomial approximation of a modified exponential evolution operator, had been developed already for this type of problems where G is a linear, time independent matrix and s is a constant vector. In this paper we will describe a new algorithm for the more general case where s is a time-dependent r.h.s vector. An iterative version of the new algorithm can be applied to the general case where G depends on t or u. Numerical results for Schrödinger equation with time-dependent potential and to non-linear Schrödinger equation will be presented.

Keywords Time-dependent pde's · System of ode's · Propagator · Evolution operator · Schrödinger

H. Tal-Ezer (✉)
School of Computer Sciences, Academic College of Tel-Aviv Yaffo, Rabenu Yeruham St., Tel-Aviv 61803, Israel
e-mail: hillel@mta.ac.il

R. Kosloff · I. Schaefer
Institute of Chemistry and The Fritz Haber Research Center, The Hebrew University, Jerusalem 91904, Israel

R. Kosloff
e-mail: ronnie@fh.huji.ac.il

I. Schaefer
e-mail: ido.schaefer@mail.huji.ac

Published online: 07 March 2012

1 Introduction

The time dependent Schrödinger equation is of fundamental importance, it governs quantum dynamics. As a result any simulation of quantum phenomena requires an effective scheme to represent and solve this equation:

$$i\psi_t = H\psi \tag{1.1}$$

where ψ is a vector representing the wave function and H the Hamiltonian operator [18]. Applications differ considerably. The dimension of Hilbert space required to represent the wave function ψ can vary from 2, for a two-level-system, to $\sim 2^{30}$ in practical applications. When the size of Hilbert space becomes too large to be represented directly, approximate methods are employed which lead to a nonlinear version of the Schrödinger equation [12].

The central role of (1.1) in quantum dynamical simulations has generated a wealth of numerical methods to solve the equation. For low dimensions, the common approach is based on diagonalizing the Hamiltonian operator H. For higher dimensions, this becomes impractical and one has to resort to matrix-free methods which require only the evaluation of the operation of the Hamiltonian on a vector. As a result, an implicit knowledge of the Hamiltonian is sufficient.

Many methods have been developed and implemented to propagate the equation in time. Typically the propagation period is divided into time steps. As a result the error in each time step will accumulate. This means that effective methods should have as large as possible time step and have a high accuracy within a time step. For time independent Hamiltonian operators a global polynomial expansion of the propagator is the method of choice [3]:

$$\psi(t) = e^{-iHt}\psi(0) \approx \sum_n a_n(t) P_n(H)\psi(0) \tag{1.2}$$

where $P_n(x)$ is a polynomial of order n which is evaluated recursively. The most popular choice has been the Chebychev polynomial [15] due to its exponential rate of convergence. Other polynomials have been tried with similar or inferior results.

In many applications, the Hamiltonian is explicitly time dependent. These include systems subject to a time dependent electromagnetic field (spectroscopy), quantum control which requires to infer the time dependent field that leads to a desired outcome such as a quantum gate [7]. In these problems the remedy to overcome the explicit time dependence was to employ a short time step in which the field is approximated as piecewise constant. This solution immediately degrades the accuracy to first order in the time step. Four general approached have been explored to overcome this difficulty.

1. Solving the equations using general Taylor based solvers such as Runge Kutta or second order differencing. These methods have slow convergence properties [8].
2. Employing the (t, t') method which eliminates the explicit time dependence by embedding the problem in a larger Hilbert space adding time translation to the Hamiltonian $H(t) \to H(t') + i\frac{\partial}{\partial t'}$. The method restores the accuracy of the high order polynomial expansion but has been found to be expensive in use [13].
3. Another class of approaches rely on the Magnus expansion to overcome the problem of time ordering [17]. The solution is cast into the form: $\psi(t) = e^U\psi(0)$ and approximated as $e^U \approx e^{A_1}e^{A_2}e^{A_3}e^{A}\cdots$. This type of solution includes the split operator method [14] as well as polynomial approximations of the exponent [1].
4. When the Hamiltonian can be split as: $H = H_0 + V(t)$ then $\psi(t) = e^{-iH_0t}\psi(0) + \int_0^t e^{-iH_0(t-t')}V(t')\psi(t')dt'$. This formal solution establishes the base for a polynomial approximation of the result [10, 11].

When considering nonlinear version of the Schrödinger equation, such as the Gross–Pitaevskii equation or time dependent density functional equations, methods 2 and 3 are not applicable and we are left with options based on 1 and 4. The new algorithm presented in this paper belongs to the fourth approach. We will demonstrate that the new algorithm is highly efficient with respect to accuracy versus numerical effort, both for linear time dependent problems as well as for non linear versions of the Schrödinger equation.

2 The New Algorithm (Linear Case)

Let us consider a general system of ode's of the form

$$u_t = Gu + s, \tag{2.1}$$

$$u(0) = v_0, \tag{2.2}$$

where G is a constant, $N \times N$ matrix. If s is constant then, by Duhamel principle, the solution is

$$u(t) = e^{Gt}v_0 + \int_0^t e^{G(t-\tau)}s\,d\tau. \tag{2.3}$$

Formal integration results in

$$u(t) = e^{Gt}v_0 + f_1(G,t)s, \tag{2.4}$$

where

$$f_1(z,t) = \begin{cases} \frac{1}{z}(e^{zt}-1) & z \neq 0 \\ t & z = 0. \end{cases} \tag{2.5}$$

Since

$$e^{zt} = zf_1(z,t) + 1, \tag{2.6}$$

then

$$e^{Gt} = Gf_1(G,t) + I \tag{2.7}$$

and therefore

$$u(t) = v_0 + f_1(G,t)v_1 \tag{2.8}$$

where $v_1 = Gv_0 + s$.

Going one step further, let us consider the system

$$u_t = Gu + s_0 + ts_1. \tag{2.9}$$

Using similar steps as above, we get the formal solution

$$u(t) = v_0 + tv_1 + f_2(G,t)v_2, \tag{2.10}$$

where

$$f_2(z,t) = \begin{cases} \frac{1}{z^2}(e^{zt}-1-zt) & z \neq 0 \\ \frac{t^2}{2} & z = 0 \end{cases} \tag{2.11}$$

and $v_2 = Gv_1 + s_1$. The following lemma applies to the general case.

Lemma *The formal solution of the set of ode's*

$$u_t = Gu + \sum_{j=0}^{m-1} \frac{t^j}{j!} s_j \tag{2.12}$$

is

$$u = \sum_{j=0}^{m-1} \frac{t^j}{j!} v_j + f_m(G,t) v_m, \tag{2.13}$$

where v_j satisfy the recurrence relation

$$v_0 = u_0 \tag{2.14}$$

$$v_j = G v_{j-1} + s_{j-1} \quad 1 \le j \le m \tag{2.15}$$

and

$$f_m(z,t) = \begin{cases} \frac{1}{z^m}(e^{zt} - \sum_{j=0}^{m-1} \frac{(zt)^j}{j!}) & z \neq 0 \\ \frac{t^m}{m!} & z = 0. \end{cases} \tag{2.16}$$

Proof It is easily verified that

$$\frac{df_m}{dt} = z f_m + \frac{t^{m-1}}{(m-1)!}. \tag{2.17}$$

Hence

$$u_t = \sum_{j=0}^{m-2} \frac{t^j}{j!} v_{j+1} + G f_m v_m + \frac{t^{m-1}}{(m-1)!} v_m \tag{2.18}$$

or

$$u_t = \sum_{j=0}^{m-1} \frac{t^j}{j!} v_{j+1} + G f_m v_m. \tag{2.19}$$

Using (2.15) we get

$$u_t = G \sum_{j=0}^{m-1} \frac{t^j}{j!} v_j + \sum_{j=0}^{m-1} \frac{t^j}{j!} s_j + G f_m v_m. \tag{2.20}$$

Hence

$$u_t = G\left(\sum_{j=0}^{m-1} \frac{t^j}{j!} v_j + f_m v_m \right) + \sum_{j=0}^{m-1} \frac{t^j}{j!} s_j \tag{2.21}$$

or

$$u_t = Gu + \sum_{j=0}^{m-1} \frac{t^j}{j!} s_j \tag{2.22}$$

and the proof is concluded. □

Remark When z is very small, computing $f_m(z,t)$ as defined in (2.16) can be unstable due to roundoff errors. Possible remedy is to use instead an approximation based on Taylor expansion

$$f_m(z,t) = t^m \sum_{j=0}^{\infty} \frac{(zt)^j}{(m+j)!}. \tag{2.23}$$

The solution vector u can be approximated with high accuracy as

$$u \approx \sum_{j=0}^{m-1} \frac{t^j}{j!} v_j + p_k(G,t)v_m \tag{2.24}$$

where $p_k(z,t)$ is 'optimal' polynomial which approximates $f_m(z,t)$ where $z \in D$ and D is a domain in the complex plane which includes all the eigenvalues of G. The p_k polynomial can be based on Chebyshev expansion [15], Arnoldi approach [4, 16] or Newton interpolation approach [17].

In the more general case where s is any function of t we do first Chebyshev approximation of s

$$s(t) \approx \sum_{j=0}^{m-1} \tilde{s}_j T_j(t) \tag{2.25}$$

and then transform the expansion to the Taylor-like representation as in (2.12) [11].

3 Time-Dependant G

Let us consider now the case where the matrix G depends on t

$$u_t = G(t)u + s(t), \qquad u^0 = u(0), \quad 0 \le t \le T. \tag{3.1}$$

(The time dependent Schrödinger equation where the potential depends on t is an example of such an equation.)

In order to apply the new algorithm in this case, one has to resort to a time-steps algorithm. Consider that we have marched already to time level t_n and we want to compute the solution at time level t_{n+1}. (3.1) can be written as

$$u_t = G_n u + s_n(t), \quad 0 \le t \le T \tag{3.2}$$

where

$$G_n = G\left(t_n + \frac{\Delta T}{2}\right), \qquad s_n(t) = s(t) + \big(G(t) - G_n\big)u, \tag{3.3}$$

and

$$\Delta t = t_{n+1} - t_n. \tag{3.4}$$

Observe that $s_n(t)$ depends on u which is unknown yet at the time interval $[t_n, t_n + \Delta t]$ but, as described in Main Algorithm below, a set of approximated vectors u_n^j, $j = 1, \ldots, m$ which approximate the solution at the Chebyshev time points

$$t_j = t_n + \frac{\Delta t}{2}(1 - y_j), \qquad y_j = \cos\left(\frac{(j-1)\pi}{m-1}\right), \quad 1 \le i \le m, \tag{3.5}$$

can be computed in the previous time step and is used to compute the s_j vectors as defined in (2.12). Only in the first step one has to use an iterative algorithm in order to compute the set of approximated solution vectors at the time points

$$t_j = \frac{\Delta t}{2}(1 - y_j), \qquad y_j = \cos\left(\frac{(j-1)\pi}{m-1}\right), \quad 1 \le j \le m \tag{3.6}$$

where the first guess is

$$u_j^1 = u^0, \quad 1 \le j \le m. \tag{3.7}$$

The iterative algorithm is stopped when $\|u_m^{k+1} - u_m^k\|$ satisfies the desired accuracy.

First Step Algorithm Given: $u^0,\ \epsilon,\ m$ and let $t_j = \frac{\Delta t}{2}(1 - \cos(\frac{(j-1)\pi}{m-1})),\ 1 \le j \le m$

1. $u_j = u^0,\ j = 1, \ldots, m$
2. Compute $\hat{s}_j = s_0(t_j)$ (defined in (3.3))
3. Compute s_j (defined in (2.12)) by using cosine transform of $\hat{s}_j$ and then Taylor-like transform
4. Use (2.24) to compute $u_j^{\text{new}},\ 1 \le j \le m$
5. if $\|u_m^{\text{new}} - u_m\| \le \epsilon$ then stop
6. $u_j = u_j^{\text{new}}, 1 \le j \le m$
7. go to 2

After computing the initial solution vectors at the time points $t_j = \frac{\Delta t}{2}(1 - \cos(\frac{(j-1)\pi}{m-1}))$, $1 \le j \le m$ we are ready to continue with the main algorithm which computes the solution at the time interval $[0, T]$.

Main Algorithm Given: $v_0,\ m,\ \{u_j\}_{j=1}^m,\ T, t = 0, n = 0$

1. Let $t_j^1 = t + \frac{\Delta t}{2}(1 - \cos(\frac{(j-1)\pi}{m-1})),\ t_j^2 = t + \Delta t + \frac{\Delta t}{2}(1 - \cos(\frac{(j-1)\pi}{m-1})), 1 \le j \le m$
2. Compute $\hat{s}_j = s_n(t_j^1)$, (defined in (3.3))
3. Compute s_j (defined in (2.12)) by using cosine transform of $\hat{s}_j$ and then Taylor-like transform
4. Use (2.24) to compute $\{u_j\}_{j=1}^m$ at $\{t_j^2\}_{j=1}^m$
5. if $t = T$ then stop
6. $t = t + \Delta t,\ n = n + 1$
7. go to 1

Observe that $t_m^1 = t_1^2 = t + \Delta t$ hence, at each step, the solution vector at this point is computed twice. The first one is the predictor and the second one is the corrector.

4 Nonlinearity

Let us consider now the nonlinear case.

$$u_t = G(u)u + s(u), \quad 0 \le t \le T. \tag{4.1}$$

Implementation of the new algorithm in this case is almost the same as it is done in the case described in the previous section. Equation (4.1) can be written as

$$u_t = G_n u + s_n, \quad 0 \le t \le T \tag{4.2}$$

where

$$G_n = G\left(u\left(t_n + \frac{\Delta T}{2}\right)\right), \qquad s_n = s(u) + \big(G(u) - G_n\big)u. \tag{4.3}$$

The rest of the description of the algorithm is exactly the same as in the previous section.

5 Numerical Examples

The numerical examples presented in this section address the case where the eigenvalues of the spatial matrix G are on the imaginary axis. In this case one can use the Chebyshev approach [15]. In a future paper we will treat the more general case (e.g. boundary value problems, advection diffusion) where the domain of eigenvalues is on the left side of the complex plane.

Example 1 (Time-dependent r.h.s.) Let us consider the differential equation

$$u_t = u_x + s(x,t) \quad 0 \le x \le 2\pi \tag{5.1}$$

where

$$s(x,t) = \sin(6x)\cos(t) - 2\cos(10x)\cos(2t) - 6\cos(6x)\sin(t) + 10\sin(10x)\sin(2t). \tag{5.2}$$

The exact solution is

$$u(x,t) = \sin(t)\sin(6x) + \sin(2t)\cos(10x). \tag{5.3}$$

Since we have periodicity in space we can use spectral Fourier for space approximation. It results in a set of ode's

$$u_t = Gu + s \tag{5.4}$$

where u is a vector of length n (number of grid points), G is an $n \times n$ matrix which carries out the Fourier spectral differentiation and s is a vector of length n which is time-dependent.

We have solved this problem in the time interval $[0, 5]$. Since the solution is periodic with highest mode equal to 10, using $n = 32$ is suffice to compute exactly the spatial derivative. Hence, the error comes solely from time approximation.

In order to compute solution in this time interval which satisfies

$$\|u - u_{exact}\| \le 10^{-5} \tag{5.5}$$

we had to use $m = k = 14$ (these parameters are defined in (2.12) and (2.24)). It means that all together we had to do 28 matrix-vector multiplications.

Applying standard ODE45 for this problem, we had to do 860 matrix-vector multiplications in order to compute the solution to the desired accuracy.

Example 2 (Schrödinger equation with time-dependent potential) As a second example, we consider harmonic oscillator of mass $m = 1$ and frequency $\omega = 1$ driven by a linearly polarized electromagnetic field with frequency $\nu = 1$. We have to solve

$$\psi_t = -iH(r,t)\psi \tag{5.6}$$

where the time-dependent Hamiltonian is given by

$$H(r,t) = -\frac{1}{2}\frac{\partial^2}{\partial r^2} + \frac{1}{2}r^2 + r\sin^2\left(\frac{\pi t}{T}\right)\cos(t). \tag{5.7}$$

Table 1 RK4

Time-steps	Matvecs	Relative L2 error
1500	6000	5.6e-04
3000	12000	3.5e-05
6000	24000	2.2e-06

Table 2 New algorithm, $m = k = 7$

Time-steps	Matvecs	Relative L2 error
350	4563	3.7e-02
400	5213	3.9e-08
600	7813	3.9e-10

Table 3 New algorithm, $m = k = 8$

Time-steps	Matvecs	Relative L2 error
300	4515	2.3e-08
400	6015	1.3e-09
450	6765	4.8e-10

Table 4 New algorithm, $m = k = 9$

Time-steps	Matvecs	Relative L2 error
280	4777	6.0e-08
350	5967	1.4e-09
400	6817	3.2e-10

The final time is set to $T = 15$. The Hamiltonian is represented on a Fourier grid with $n = 128$ grid points, and $r_{\max} = 10 = -r_{\min}$. We have used the spectral Fourier method to approximate the spatial derivatives.

Taking the initial wave function to be

$$\psi(r, 0) = e^{-r^2} \tag{5.8}$$

we computed the numerical solution by two methods:

1. RK4 (Runge-Kutta of order 4)
2. the new algorithm.

In Tables 1, 2, 3, 4, 5 and 6 matvecs represents the number of applications of the Hamiltonian.

The first table presents the RK4 results. For stability, the time step should be $\Delta t = 0.01$, hence the minimal number of time steps needed to march to $T = 15$ is 1500.

Observe that dividing the time step by 2, the error is reduced by a factor of almost 16 as it should be since RK4 is a scheme of order 4. Hence, in order to get high accuracy, e.g. of order 10^{-10}, one should do around 192000 matrix-vector multiplications.

In the next few tables we present the results for the new algorithm. The tables differ by the m and k parameters where m is the number of Chebyshev points in the interval $[t, t + \Delta t]$ and k is the degree of the polynomials used to approximate the function f_m.

Table 5 RK4

Time-steps	Matvecs	Relative L2 error
660	2640	4.96e-01
1320	5280	2.00e-02
2640	10560	1.20e-03
5280	21120	7.29e-05

Table 6 New algorithm, $m = k = 7$

Time-steps	Matvecs	Relative L2 error
300	4043	3.3e-05
500	6643	9.5e-08
700	9243	1.7e-09

Remark The minimal number of time-steps presented in the last 3 tables were such that taking smaller number will result in instability.

Observe that the new algorithm is significantly more efficient then RK4, especially when one is interested in high accuracy. In this case, the new algorithm is almost 30 times more efficient then RK4.

Example 3 (Nonlinear Schrödinger equation) For a nonlinear example we choose the Gross–Pitaevskii equation describing the dynamics of a Bose-Einstein-Condensate (BEC) in a harmonic trap:

$$\psi_t = -i\,H(r,\psi)\psi \tag{5.9}$$

where the Hamiltonian is given by

$$H(r,\psi) = -\frac{1}{2}\frac{\partial^2}{\partial r^2} + \frac{1}{2}r^2 + |\psi|^2. \tag{5.10}$$

The final time is set to $T = 10$. The Hamiltonian is represented on a Fourier grid with $n = 128$ grid points, and $r_{\max} = 8\sqrt{\pi} = -r_{\min}$. The spectral Fourier method is used to approximate the spatial derivatives.

The initial state is

$$\psi_0 = e^{i8r} v_0 \tag{5.11}$$

where v_0 is the eigenvector related to the smallest eigenvalue of the nonlinear Hamiltonian.

As in the previous example, we computed the numerical solution by RK4 and by the new algorithm.

Table 5 presents the RK4 results. For stability, the time step should be $\Delta t = 0.01515$, hence the minimal time steps needed to march to $T = 10$ is 660.

Taking into account that RK4 is a scheme of order 4 we can conclude that in order to get high accuracy, e.g. of order 10^{-10}, one should do around 382000 matrix-vector multiplications.

In Tables 6 and 7 we present the results for the new algorithm. As in the previous example, the tables differ by the m and k parameters.

Observe that for moderate accuracy of order 10^{-5}, 21120 matvecs were needed in the RK4 case while using the new algorithm with $m = k = 9$, only 3587 matvecs were needed. The increase in efficiency is more pronounced when high accuracy is needed. For order of

Table 7 New algorithm, $m = k = 9$

Time-steps	Matvecs	Relative L2 error
200	3587	5.67e-05
300	5287	1.14e-07
400	6987	3.00e-09
500	8687	6.43e-10

10^{-10} accuracy, 382000 matvecs are needed in the RK4 case compared to 8687 matvecs for the new algorithm.

6 Conclusions

In this paper we have presented a new algorithm for solving a class of linear and nonlinear Schrödinger equations which can be applied to general system of ode's. In the stationary linear case it is possible to reach the upper time level in one step with very high accuracy. Due to the fact that there is only one step, the accuracy is not deteriorating since there is no accumulation of errors. In the case where the matrix involved depends on time or in the case of nonlinearity, the time interval should be divided to time steps but the size of the time step is significantly larger than what is needed in standard explicit algorithms like Runge-Kutta.

The high accuracy (spectral) of the algorithm can be traced to the fact that the algorithm does not use any Taylor considerations. Taylor theorem is an extremely important tool in analysis but due to its locality it can lead to inferior numerical approximation. We believe that whenever it is possible to develop an algorithm which is Taylor free, one should explore this possibility.

7 Remarks

During the refereeing process we came to know of methods known as Exponential Integrators (e.g. [2, 5, 6, 9]) which also make use of the functions defined in (2.16). The algorithms described in those paper are, like Runge-Kutta approach, based on Taylor considerations while the algorithm described here is Taylor-free. This is the main difference between the two approaches. Since the present algorithm is Taylor-free, there is no meaning to the term—"order of the method" which we have in the Exponential Integrator methods.

References

1. Alvermanna, A., Fehskeb, H.: High-order commutator-free exponential time-propagation of driven quantum systems. J. Comput. Phys. **230**(15), 5930–5956 (2011)
2. Caliari, M., Ostermann, A.: Implementation of exponential Rosenbrock-type integrators. Appl. Numer. Math. Arch. **59**(3–4), 568–581 (2009)
3. Leforestier, C., Bisseling, R., Cerjan, C., Feit, M., Friesner, R., Guldberg, A., Dell Hammerich, A., Julicard, G., Karrlein, W., Dieter Meyer, H., Lipkin, N., Roncero, O., Kosloff, R.: A comparison of different propagation schemes for the time dependent Schrödinger equation. J. Comput. Phys. **94**, 59–80 (1991)
4. Higham, N.J., Al-Mohy, A.H.: Computing matrix functions. Acta Numer. **19**, 159–208 (2010)
5. Hochbruck, M., Ostermann, A.: Explicit exponential Runge-Kutta methods for semilinear parabolic problems. SIAM J. Numer. Anal. **43**, 1069–1090 (2005)

6. Hochbruck, M., Ostermann, A.: Exponential integrators of Rosenbrock-type. Oberwolfach Rep. **3**, 1107–1110 (2006)
7. Palao, J.P., Kosloff, R.: Quantum computing by an optimal control algorithm for unitary transformations. Phys. Rev. Lett. **89**, 188301 (2002)
8. Kosloff, D., Kosloff, R.: A Fourier method solution for the time dependent Schrödinger equation as a tool in molecular dynamics. J. Comput. Phys. **52**, 35–53 (1983)
9. Hochbruck, M., Lubich, C.: Exponential integrators for quantum-classical molecular dynamics. BIT **39**, 620–645 (1999)
10. Ndong, M., Tal-Ezer, H., Kosloff, R., Koch, C.: A Chebychev propagator with iterative time ordering for explicitly time-dependent Hamiltonians. J. Chem. Phys. **132**, 064105 (2010)
11. Ndong, M., Tal-Ezer, H., Kosloff, R., Koch, C.: A Chebychev propagator for inhomogeneous Schrödinger equation. J. Chem. Phys. **130**, 124108 (2009)
12. Meyer, H.-D., Manthe, U., Cederbaum, L.S.: The multi-configurational time-dependent Hartree approach. Chem. Phys. Lett. **165**, 73 (1990)
13. Peskin, U., Kosloff, R., Moiseyev, N.: The solution of the time dependent Schrödinger equation by the (t, t') method: the use of global polynomial propagators for time dependent Hamiltonians. J. Chem. Phys. **100**, 8849–8855 (1994)
14. Feit, M.D., Fleck, J.A. Jr., Steiger, A.: Solution of the Schrödinger equation by a spectral method. J. Comput. Phys. **47**, 412 (1982)
15. Tal Ezer, H., Kosloff, R.: An accurate and efficient scheme for propagating the time dependent Schrödinger equation. J. Chem. Phys. **81**, 3967–3970 (1984)
16. Tal Ezer, H.: On restart and error estimation for Krylov approximation of $w = f(A)v$. SIAM J. Sci. Comput. **29**(6), 2426–2441 (2007)
17. Tal Ezer, H., Kosloff, R., Cerjan, C.: Low order polynomial approximation of propagators for the time dependent Schrödinger equation. J. Comput. Phys. **100**, 179–187 (1992)
18. Tannor, D.J.: Introduction to Quantum Mechanics: A Time-Dependent Perspective. University Science Press, Sausalito (2007)