Journal of Computational Physics 303 (2015) 396–412

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An efficient algorithm based on splitting for the time integration of the Schrödinger equation

Sergio Blanes [a], Fernando Casas [b,*], Ander Murua [c]

[a] Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, E-46022 Valencia, Spain
[b] Institut de Matemàtiques i Aplicacions de Castelló and Departament de Matemàtiques, Universitat Jaume I, E-12071 Castellón, Spain
[c] Konputazio Zientziak eta A.A. Saila, Informatika Fakultatea, UPV/EHU, Donostia/San Sebastián, Spain

ARTICLE INFO

Article history:
Received 23 February 2015
Received in revised form 28 September 2015
Accepted 28 September 2015
Available online 9 October 2015

Keywords:
Symplectic splitting methods
Time-dependent Schrödinger equation
Error analysis

ABSTRACT

We present a practical algorithm based on symplectic splitting methods intended for the numerical integration in time of the Schrödinger equation when the Hamiltonian operator is either time-independent or changes slowly with time. In the later case, the evolution operator can be effectively approximated in a step-by-step manner: first divide the time integration interval in sufficiently short subintervals, and then successively solve a Schrödinger equation with a different time-independent Hamiltonian operator in each of these subintervals. When discretized in space, the Schrödinger equation with the time-independent Hamiltonian operator obtained for each time subinterval can be recast as a classical linear autonomous Hamiltonian system corresponding to a system of coupled harmonic oscillators. The particular structure of this linear system allows us to construct a set of highly efficient schemes optimized for different precision requirements and time intervals. Sharp local error bounds are obtained for the solution of the linear autonomous Hamiltonian system considered in each time subinterval. Our schemes can be considered, in this setting, as polynomial approximations to the matrix exponential in a similar way as methods based on Chebyshev and Taylor polynomials. The theoretical analysis, supported by numerical experiments performed for different time-independent Hamiltonians, indicates that the new methods are more efficient than schemes based on Chebyshev polynomials for all tolerances and time interval lengths. The algorithm we present automatically selects, for each time subinterval, the most efficient splitting scheme (among several new optimized splitting methods) for a prescribed error tolerance and given estimates of the upper and lower bounds of the eigenvalues of the discretized version of the Hamiltonian operator.

1. Introduction

When investigating the dynamical behavior of quantum systems of low to moderate dimension, very often it is necessary to solve numerically the time-dependent Schrödinger equation ($\hbar = 1$)

$$i\hbar \frac{\partial}{\partial t}\psi(x,t) = \hat{H}(t)\psi(x,t), \qquad \psi(x,0) = \psi_0(x). \tag{1}$$

[*] Corresponding author.
E-mail addresses: serblaza@imm.upv.es (S. Blanes), Fernando.Casas@uji.es (F. Casas), Ander.Murua@ehu.es (A. Murua).

http://dx.doi.org/10.1016/j.jcp.2015.09.047
0021-9991/© 2015 Elsevier Inc. All rights reserved.

Here $\hat{H}(t)$ is a time-dependent Hamiltonian operator, $\psi : \mathbb{R}^d \times \mathbb{R} \longrightarrow \mathbb{C}$ is the wave function representing the state of the system, and $\psi_0(x)$ is the initial state.

We assume that the Hamiltonian operator changes slowly with time, so that the evolution operator can be effectively approximated in a step-by-step manner. We first subdivide the time integration interval in a number of sufficiently short subintervals of length τ, and then compute approximations $\psi_k = \psi(\cdot, t_k)$ of the wave function at times $t_k = k\tau$, $k = 1, 2, 3, \ldots$, by successively solving (1) but now with a time-independent Hamiltonian operator $\overline{H}_k$ in each of these subintervals $[t_{k-1}, t_k]$. As suggested in [10], it is natural to choose the matrix $\overline{H}_k$ corresponding to a first order Magnus approximation [20], namely,

$$\overline{H}_k = \frac{1}{\tau} \int_{t_{k-1}}^{t_k} \hat{H}(t)\, dt. \tag{2}$$

Typically, $\hat{H}(t) = \hat{T} + \hat{V}(t)$, with the kinetic energy operator $\hat{T} = -\Delta/(2\mu)$ for a reduced mass $\mu > 0$ and a time-dependent potential $\hat{V}(t)$, in which case, $\overline{H}_k = \hat{T} + \overline{V}_k$, where $\overline{V}_k$ is the average of $\hat{V}(t)$ over the interval $[t_{k-1}, t_k]$. Of course, considering higher order Magnus approximations is also possible [4], and particularly interesting if the Hamiltonian operator $\hat{H}(t)$ does not vary very slowly with time. See for instance [5,16] for other alternatives for the case of strongly time-dependent Hamiltonian operators.

When discretized in space, the Schrödinger equation with the time-independent Hamiltonian operator obtained for each time subinterval $[t_{k-1}, t_k]$ can be recast as a classical linear autonomous Hamiltonian system corresponding to a system of coupled harmonic oscillators. Several techniques can be used for the space discretization, depending on the particular problem one aims to analyze: finite difference schemes, spectral methods based on collocation with trigonometric polynomials, Galerkin method with a Hermite basis, etc., both in one or more dimensions (see [19] and references therein). The space discretization process restricts the energy range of the approximation and imposes an upper bound to the high frequency components represented by the discrete solution.

In any event, once this process has been carried out, one is led to compute $u_k \in \mathbb{C}^N$ for $k = 1, 2, 3, \ldots$, where the sequence $\{u_k\}$ now represents a fully discretized version of the wave function $\psi(x, t)$ at the time grids points t_k and the N space grid points, with N usually a large number. Here, $u_k = u(t_k)$, $k = 0, 1, 2, \ldots$, where $u(t)$ is the solution of

$$i \frac{d}{dt} u(t) = \widetilde{H}(t)\, u(t), \qquad u(0) = u_0 \in \mathbb{C}^N, \tag{3}$$

$\widetilde{H}(t)$ is a piecewise constant $N \times N$ matrix,

$$\widetilde{H}(t) = H_k \quad \text{for} \quad t \in [t_{k-1}, t_k],$$

and each H_k is a discretized counterpart of the averaged Hamiltonian operator (2). The $N \times N$ matrices H_k (and in particular its discrete spectra) depends of course on the particular space discretization carried out. We will hereafter assume that each H_k is a real symmetric matrix which implies that it can be diagonalized with real eigenvalues.

As $\widetilde{H}(t)$ is piecewise constant, the initial value problem (3) can be exactly solved in terms of matrix exponentials $e^{-i\tau H_k}$, and we are led to compute u_k for $k = 1, 2, \ldots$ as

$$u_k = e^{-i\tau H_k} u_{k-1}. \tag{4}$$

However, computing $e^{-i\tau H_k}$ by diagonalizing the constant matrix H_k (usually, of large dimension and large norm) might be exceedingly costly for some problems.

In the present work, we focus on developing efficient approximations to $e^{-i\tau H} v$ for arbitrary $N \times N$ real symmetric matrices H and arbitrary vectors $v \in \mathbb{C}^N$ by only performing matrix–vector products of the form Hv. Our final goal is to produce an efficient algorithm that computes $e^{-i\tau H} v$ for given $\tau \in \mathbb{R}$ and $v \in \mathbb{C}^N$ within a prescribed error tolerance with a minimum number of matrix–vector products Hv. This algorithm is based on symplectic splitting methods.

Of course, the error coming from the approximation to the exponential in (4) is only one of the contributions to the error of the global process of constructing the approximation $\{u_k\}$ to the solution of the original problem (1). In addition, the space discretization error and the error of the first order Magnus approximation should also be taken into account. Depending on the magnitude of the two later error contributions, one should choose more or less stringent error tolerances for the approximate computation of the expressions $e^{-i\tau H_k} u_{k-1}$ in (4).

Our approach for approximating $e^{-i\tau H} v$ is closely related to other polynomial approximations of the form

$$e^{-i\tau H} v \approx P_m(\tau H) v, \tag{5}$$

where $P_m(y)$ is a polynomial in y that approximates the exponential e^{-iy}. Here also multiplications of the matrix H with vectors v are only involved, and these can be evaluated in complex variables with the complex-to-complex Fast Fourier Transform (FFT) algorithm [7,14,15,17]. Different choices for such $P_m(y)$ are available, namely truncated Taylor or Chebyshev series expansion of e^{-iy} for an appropriate real interval of y, or a Lanczos approximation, where the polynomial is determined by a Galerkin approximation on the Krylov space spanned by $v, Hv, \ldots, H^{m-1}v$ [25].

The main difference of our procedure with respect to the polynomial approximation (5) is the following. Whereas in (5) the approximation of $\mathrm{e}^{-i\tau H}v$ is obtained by computing products of the form Hv, where $v \in \mathbb{C}^N$, with symplectic splitting methods one writes $v = q + ip$, $q, p \in \mathbb{R}^N$, and the algorithm proceeds by successively computing *real* matrix–vector products Hq and Hp with different weights. In this way, the real and imaginary parts of $\mathrm{e}^{-i\tau H}v$ are approximated in a different way, with a considerably reduced computational cost. More specifically, if a spatial discretization based on Fourier spectral methods is considered, then the cost of computing Hv, $v \in \mathbb{C}^N$, amounts essentially to one complex-to-complex FFT and its inverse, whereas in the case of Hv, $v \in \mathbb{R}^N$, one has to evaluate one real-to-complex FFT and its inverse complex-to-real FFT, and this process requires half the computing time of the fully complex case.

Although initially motivated by the integration in time of the Schrödinger equation (1), the algorithm proposed in this paper can indeed be applied to approximate $\mathrm{e}^{-i\tau H}v$ for *any* real symmetric matrix H and any complex vector v under the same assumptions as the Chebyshev method. As a matter of fact, the theoretical analysis carried out here and supported by numerical experiments performed for different real symmetric matrices H, indicates that our new symplectic splitting schemes for approximating $\mathrm{e}^{-i\tau H}v$ are more efficient than schemes based on Chebyshev polynomials for all tolerances and values of τ. The algorithm we present automatically selects the most efficient splitting scheme (among several new optimized splitting methods) for a prescribed error tolerance, values of τ, and given estimates of the upper and lower bounds of the eigenvalues of the matrix H, $E_{\min}$ and $E_{\max}$. As a result, it turns out to be between 1.4 and 2 times faster than the Chebyshev method for the same accuracy. In addition, the favorable geometric properties of symplectic splitting methods [3] result (compared to Chebyshev) in reduced energy and unitarity errors for large values of τ. The computation of the coefficients of the schemes, which constitutes a non-trivial task by itself, is largely based on the stability and error analysis of splitting methods carried out in [2,3].

In recent years several efficient numerical integration techniques for the time integration of the linear and nonlinear Schrödinger equation have been proposed and analyzed. We can mention, in particular, the so-called unitary split operator algorithms, which take advantage of the usual separation of the Hamiltonian into kinetic and potential energy, $H = T + V$, to construct compositions of the form $\mathrm{e}^{-ib_m\tau V}\,\mathrm{e}^{-ia_m\tau T}\cdots\mathrm{e}^{-ib_1\tau V}\,\mathrm{e}^{-ia_1\tau T}$, where $\{a_i, b_i\}$ are appropriately chosen real coefficients [12,29,23,30]. In the context of disordered systems, the resulting discrete nonlinear Schrödinger equation has been integrated by high order symplectic schemes in [22] and [26]. In this last reference, methods designed by the present authors for near integrable Hamiltonians [1] have been shown to be quite efficient for this problem. However, all the theoretical results concerning the error analysis of unitary split operator methods we have found in the literature, in contrast to Chebyshev based methods or symplectic splitting methods, depend on some assumptions on the smoothness of the initial state.

The plan of the paper is the following. Since our procedure, in contrast with the above mentioned techniques, may be considered as an alternative to the Chebyshev method, in Section 2 we summarize the main features of the schemes based on this polynomial approximation of $\mathrm{e}^{-i\tau H}v$. In Section 3 we perform an error analysis of symplectic splitting methods in this context, and the actual algorithm is presented in Section 4. Next, the efficiency of our algorithm for approximating $\mathrm{e}^{-i\tau H}v$ is compared with Chebyshev (and Taylor as a reference) in Section 5 on a pair of selected numerical examples involving constant matrices H.

2. Polynomial approximations

2.1. General considerations

As we have seen, to get the solution of the discretized version (3) of the Schrödinger equation (1), one has to compute $u_k = u(t_k)$ for $k = 1, 2, 3, \ldots$. This is done by successively evaluating in (4) the product of a matrix exponential $\mathrm{e}^{-i\tau H_k}$ with a vector $u_{k-1} \in \mathbb{C}^N$. Here, τ is the length of the subintervals $[t_{k-1}, t_k]$. In what follows, we focus on the approximate computation of $\mathrm{e}^{-i\tau H}v$ for given $N \times N$ real symmetric matrix H, complex vector $v \in \mathbb{C}^N$, and $\tau \in \mathbb{R}$. We will refer to τ as the *target time-step*.

Assume that $P_m(y)$ is an mth degree polynomial approximating the function e^{-iy} in some sense. Then, $\mathrm{e}^{-i\tau H}v$ can be approximated by $P_m(\tau H)\,v$, with an error that is bounded (in Euclidean norm) as

$$\begin{aligned}\|P_m(\tau H)\,v - \mathrm{e}^{-i\tau H}v\| &\le \|P_m(\tau H) - \mathrm{e}^{-i\tau H}\|\,\|v\| \\ &\le \max_{j=0,1,\ldots,N-1} |P_m(\tau E_j) - \mathrm{e}^{-i\tau E_j}|\,\|v\|\end{aligned}$$

in terms of the (real) eigenvalues $E_0, \ldots, E_{N-1}$ of H. Assuming that the spectrum $\sigma(H) = \{E_0, \ldots, E_{N-1}\}$ is contained in an interval of the form $[E_{\min}, E_{\max}]$, then

$$\|P_m(\tau H) - \mathrm{e}^{-i\tau H}\| \le \sup_{\tau E_{\min} \le y \le \tau E_{\max}} |P_m(y) - \mathrm{e}^{-iy}|.$$

There are several possibilities to estimate $E_{\max}$ and $E_{\min}$ for different classes of matrices (see e.g. [11,21,31,32]). If H can be decomposed as the sum $H = T + V$ of two symmetric matrices with known lower and upper bounds for their eigenvalues, $E_{\min}$ (resp. $E_{\max}$) can be simply obtained as the sum of the lower (resp. upper) bounds of the eigenvalues of

T and V. This happens, in particular, when the Hamiltonian operator $\hat{H} = -\Delta/(2\mu) + \hat{V}$ is discretized by spectral Fourier collocation with N Fourier modes, in which case

$$E_{\min} = \min_x V(x), \qquad E_{\max} = \frac{1}{2\mu}\frac{N^2}{4} + \max_x V(x). \tag{6}$$

In any case, once $E_{\min}$ and $E_{\max}$ have been determined, we introduce

$$\alpha = \frac{E_{\max} + E_{\min}}{2}, \qquad \beta = \frac{E_{\max} - E_{\min}}{2}, \qquad \overline{H} = H - \alpha I, \tag{7}$$

so that the spectrum of the shifted operator $\overline{H}$ is contained in an interval centered at the origin, $\sigma(\overline{H}) = \{E_0 - \alpha, \dots, E_{N-1} - \alpha\} \subset [-\beta, \beta]$. We thus have

$$\mathrm{e}^{-i\tau H} v = \mathrm{e}^{-i\tau\alpha}\,\mathrm{e}^{-i\tau\overline{H}} v. \tag{8}$$

Hence, we will hereafter assume without loss of generality that our problem consists in approximating $\mathrm{e}^{-i\tau H}v$ for a real symmetric matrix H with $\sigma(H) \subset [-\beta, \beta]$. In that case,

$$\|P_m(\tau H) - \mathrm{e}^{-i\tau H}\| \le \epsilon_m(\beta\tau) \tag{9}$$

where

$$\epsilon_m(\theta) \equiv \sup_{-\theta \le y \le \theta} |\mathrm{e}^{-iy} - P_m(y)|. \tag{10}$$

2.2. Taylor polynomial approximation

The mth degree Taylor polynomial $P_m^T(y)$ corresponding to e^{-iy} is of course

$$P_m^T(y) \equiv \sum_{k=0}^{m} \frac{(-i)^k}{k!} y^k, \tag{11}$$

and Horner's algorithm provides an efficient way to compute the approximation $v^* = P_m^T(\tau H)v$ of $\mathrm{e}^{-i\tau H}v$, namely

```
$y_0 = v$
do $k = 1, m$
  $y_k = v - i\frac{\tau}{m+1-k} H y_{k-1}$
enddo
$v^* = y_m.$
```
(12)

The process requires storing three complex vectors (or equivalently, 6 real vectors).

An error estimate of the form (9) can be obtained with $\epsilon_m(\theta)$ in (10) replaced by its upper bound

$$\epsilon_m^T(\theta) \equiv \frac{\theta^{m+1}}{(m+1)!}. \tag{13}$$

Since $m! \sim \sqrt{2\pi m}\,(me)^m$ for large values of m [24], we can write

$$\epsilon_m^T(\theta) \sim \frac{1}{\mathrm{e}\sqrt{2\pi m}} \left(\frac{\theta\,\mathrm{e}}{m}\right)^{m+1}.$$

In consequence, we cannot expect to have a reasonably accurate approximation $P_m^T(\tau H)v$ of $\mathrm{e}^{-i\tau H}v$ unless

$$m > \mathrm{e}\,\theta = \mathrm{e}\,\beta\,\tau.$$

In other words, increasing the value of the target time-step τ and/or refining the spatial discretization (so that β gets larger) requires evaluating a higher degree Taylor polynomial.

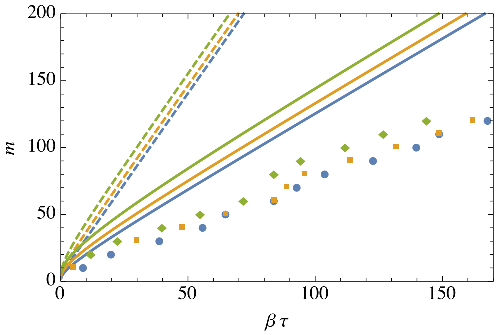

Fig. 1. Comparison of the required minimum polynomial degree m as function of $\theta = \beta\tau$ for Taylor (dashed line) and Chebyshev (continuous line) for different values of error tolerance: $\texttt{tol} = 10^{-4}, 2 \times 10^{-7}, 10^{-11}$. Diamonds, squares and circles stand for the computational cost (equivalent to a polynomial approximation of degree m) for error tolerances below 10^{-4}, 2×10^{-7} and 10^{-11}, respectively, obtained with symplectic splitting schemes in Table 1.

2.3. Chebyshev polynomial approximation

The Chebyshev polynomial expansion scheme, proposed for the first time in the context of the Schrödinger equation in [27], constitutes a standard tool to approximately compute (4). A detailed analysis of the procedure, including error estimates for the problem at hand, can be found in [19]. For completeness, we review here some of its main features.

The mth degree truncation of the Chebyshev series expansion of e^{-iy} in the interval $y \in [-\theta, \theta]$ is given by

$$P^C_{m,\theta}(y) \equiv J_0(\theta) + 2\sum_{k=1}^{m}(-i)^k J_k(\theta)\, T_k(y/\theta), \tag{14}$$

where for each k, $J_k(t)$ is the Bessel function of the first kind [24] and $T_k(x)$ is the kth Chebyshev polynomial generated from the recursion

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \qquad k \geq 1 \tag{15}$$

and $T_0(x) = 1,\ T_1(x) = x$. According with the analysis in [19], $\mathrm{e}^{-i\tau H}v$ can be approximated by $P^C_{m,\beta\tau}(\tau H)v$ with an error estimate of the form (9), where $\epsilon_m(\theta)$ in (10) is replaced by its upper bound

$$\epsilon^C_m(\theta) \equiv 4\left(\mathrm{e}^{1-\theta^2/(2m+2)^2}\frac{\theta}{2m+2}\right)^{m+1}. \tag{16}$$

In Fig. 1 we depict the minimum degree m as a function of $\theta = \beta\tau$ of Chebyshev approximations for prescribed tolerances $\texttt{tol} = 10^{-4}, 2 \times 10^{-7}, 10^{-11}$, so that $\epsilon^C_m(\beta\tau) \leq \texttt{tol}$ (continuous lines) in comparison with the corresponding degree m for Taylor approximations (dashed lines) such that $\epsilon^T_m(\beta\tau) \leq \texttt{tol}$. Notice that Chebyshev always gives a similar accuracy with a lower degree polynomial (hence, with less computational cost), with a gain in efficiency of up to a factor of two for sufficiently large values of $\theta = \beta\tau$.

Once the degree of the polynomial m has been chosen, given a certain error tolerance, target time-step τ, and bound β of $\sigma(H)$, one has to compute $P^C_{m,\beta\tau}(\tau H)\,v$ as efficiently as possible. This can be done with the Clenshaw recursive algorithm as follows: first evaluate the coefficients $c_k = (-1)^k J_k(\beta\tau)$ for $k = 0, 1, \ldots, m$ and then compute recursively

```
$d_{m+2} = 0, \qquad d_{m+1} = 0$
do $k = m, m-1, \ldots, 1, 0$
    $d_k = c_k\, v + \frac{2}{\beta} H d_{k+1} - d_{k+2}$
enddo
$v^* = d_0 - d_2,$
```
(17)

which produces $v^* \equiv P^C_{m,\beta\tau}(\tau H)\, v \approx \mathrm{e}^{-i\tau H}v$ as output. Clenshaw algorithm keeps only four complex vectors in memory,[1] but the whole procedure has to be carried out for each value of m. Since the coefficients c_k are relatively small as k grows, the Clenshaw algorithm is stable and so it is possible to work with polynomials of very high degree (even in the thousands) provided the Bessel functions are accurately computed.

[1] If the vectors are written in their real and imaginary part, and the algorithm is carried out in real variables, then the algorithm needs to store only seven real vectors instead of eight.

3. Symplectic splitting methods

3.1. General considerations

An alternative to Chebyshev polynomial approximations of $\mathrm{e}^{-i\tau H}v$ first considered in [9,10] consists in applying specially designed splitting methods to numerically integrate the system

$$i\frac{d}{dt}u = Hu, \tag{18}$$

recast in a more suitable form: By considering

$$z = \begin{pmatrix} \mathrm{Re}(u) \\ \mathrm{Im}(u) \end{pmatrix} \in \mathbb{R}^{2N},$$

equation (18) is equivalent to

$$\frac{d}{dt}z = (A+B)z, \tag{19}$$

where

$$A = \begin{pmatrix} 0 & H \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 \\ -H & 0 \end{pmatrix}. \tag{20}$$

Thus, computing $v^* = \mathrm{e}^{-i\tau H}v$, where $v = q + ip$ and $v^* = q^* + ip^*$, is equivalent to evaluating

$$\begin{pmatrix} q^* \\ p^* \end{pmatrix} = \mathrm{e}^{\tau(A+B)} \begin{pmatrix} q \\ p \end{pmatrix}.$$

The matrix exponential $\mathrm{e}^{\tau(A+B)}$ can be written as $O(\tau H)$ in terms of the orthogonal and symplectic matrix

$$O(y) = \begin{pmatrix} \cos(y) & \sin(y) \\ -\sin(y) & \cos(y) \end{pmatrix}. \tag{21}$$

To introduce general symplectic splitting methods in this setting, let us first show how the well known Strang splitting can be used to approximate $\mathrm{e}^{-i\tau H}v$. Let m be a sufficiently large positive integer, so that for $\Delta\tau = \tau/m$, we consider the approximation

$$\mathrm{e}^{\Delta\tau(A+B)} \approx \mathrm{e}^{\frac{\Delta\tau}{2}A}\,\mathrm{e}^{\Delta\tau B}\,\mathrm{e}^{\frac{\Delta\tau}{2}A}.$$

It is then clear that

$$\begin{aligned}
\mathrm{e}^{\tau(A+B)} &= \left(\mathrm{e}^{\Delta\tau(A+B)}\right)^m \\
&\approx \left(\mathrm{e}^{\frac{\Delta\tau}{2}A}\,\mathrm{e}^{\Delta\tau B}\,\mathrm{e}^{\frac{\Delta\tau}{2}A}\right)^m \\
&= \mathrm{e}^{\frac{\Delta\tau}{2}A}\left(\mathrm{e}^{\Delta\tau B}\,\mathrm{e}^{\Delta\tau A}\right)^{m-1}\,\mathrm{e}^{\Delta\tau B}\,\mathrm{e}^{\frac{\Delta\tau}{2}A},
\end{aligned}$$

or equivalently,

$$O(\tau H) = \mathrm{e}^{\tau(A+B)} \approx K(\tau H) = \mathrm{e}^{\tau a_{m+1}A}\,\mathrm{e}^{\tau b_m B}\,\mathrm{e}^{\tau a_m A}\,\cdots\,\mathrm{e}^{\tau b_1 B}\,\mathrm{e}^{\tau a_1 A}, \tag{22}$$

with

$$(a_1, b_1, a_2, \ldots, a_m, b_m, a_{m+1}) = \left(\frac{1}{2m}, \frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}, \frac{1}{m}, \frac{1}{2m}\right). \tag{23}$$

Due to the nilpotent structure of the matrices A and B in (20), the exponentials in the definition (22) of $K(\tau H)$ take a particularly simple form, namely

$$\mathrm{e}^{\tau a_j A} = \begin{pmatrix} I & a_j\tau H \\ 0 & I \end{pmatrix}, \qquad \mathrm{e}^{\tau b_j B} = \begin{pmatrix} I & 0 \\ -b_j\tau H & I \end{pmatrix}. \tag{24}$$

This analysis shows that the approximation (22) can be used to approximately compute $v^* = \mathrm{e}^{-i\tau H}v$ with the following procedure, similar in nature and equivalent in computing time to the Horner (12) and Clenshaw (17) algorithms: Given $v \in \mathbb{C}^N$,

```
$q := \mathrm{Re}(v)$,
$p := \mathrm{Im}(v)$,
do $k = 1, m$
    $q := q + a_k\, \tau\, H\, p$
    $p := p - b_k\, \tau\, H\, q$                                        (25)
enddo
    $q := q + a_{m+1}\, \tau\, H\, p$
    $v^* := q + ip$,
```

producing $v^* \approx \mathrm{e}^{-i\,\tau\,H} v$ as output. Notice that it only requires storing three real vectors of dimension N (namely q, p, and $w = Hp$ or $w = Hq$) instead of seven real vectors for the Clenshaw algorithm and six real vectors for the Horner algorithm. It is worth remarking that, since e^A and e^B are symplectic matrices, $K(\tau\, H)$ is also symplectic.

In practice, and in the same way as other polynomial approximations, it is convenient to apply Algorithm (25) with the original H replaced by the shifted version $\overline{H}$ considered in (7) (and then make use of the equality (8)), so that the spectrum of $\overline{H}$ is contained in an interval of the form $[-\beta, \beta]$ with β as sharp as possible. Therefore, in what follows we always assume that $\sigma(H) \subset [-\beta, \beta]$.

Although Algorithm (25) with coefficients (23) can be used in principle to approximate $\mathrm{e}^{-i\,\tau\,H} v$, we next show that, for given values of m and $\theta = \beta\tau$, much better approximations can be obtained if other sequences of coefficients $(a_1, b_1, a_2, \ldots, a_m, b_m, a_{m+1})$ are chosen instead. To see how this can be done, an error estimate of the corresponding approximation (22) is necessary first.

3.2. Error analysis

For a given finite sequence of real numbers

$$(a_1, b_1, a_2, \ldots, a_m, b_m, a_{m+1}), \tag{26}$$

Algorithm (25) produces an approximation of the form

$$\begin{pmatrix} q^* \\ p^* \end{pmatrix} = K(\tau\, H) \begin{pmatrix} q \\ p \end{pmatrix} \approx \mathrm{e}^{\tau\,(A+B)} \begin{pmatrix} q \\ p \end{pmatrix},$$

or equivalently, $q^* + i\,p^* \approx \mathrm{e}^{-i\,\tau\,H}(q + i\,p)$, with

$$K(\tau\, H) = \begin{pmatrix} K_{11}(\tau\, H) & K_{12}(\tau\, H) \\ K_{21}(\tau\, H) & K_{22}(\tau\, H) \end{pmatrix}. \tag{27}$$

Here $K_{11}(y)$, $K_{22}(y)$ are even polynomials of degree $2m$, $K_{12}(y)$ and $K_{21}(y)$ are odd polynomials of degree $2m - 1$ and $2m + 1$ respectively, and $\det K(y) = K_{11}(y)K_{22}(y) - K_{12}(y)K_{21}(y) \equiv 1$. It is important to remark that for a given positive integer m, compared to Horner's (12) and Clenshaw's (17) algorithms, the degree of the polynomials involved in an m-stage splitting method (26) is *twice* the degree of the corresponding Taylor and Chebyshev polynomials, with the same computational cost.

3.2.1. Error estimates for a single application of a splitting method

We next focus on obtaining upper bounds for the error

$$\begin{aligned} \|(q^* + i\,p^*) - \mathrm{e}^{-i\,\tau\,H}(q + i\,p)\| &= \left\| K(\tau\, H) \begin{pmatrix} q \\ p \end{pmatrix} - O(\tau\, H) \begin{pmatrix} q \\ p \end{pmatrix} \right\| \\ &\leq \|K(\tau\, H) - O(\tau\, H)\|\ \|q + i\,p\| \end{aligned}$$

in Euclidean norm. Since H is assumed to be a real symmetric matrix, it can be diagonalized as

$$H = P^T \begin{pmatrix} E_0 & 0 & \cdots & 0 \\ 0 & E_1 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & E_{N-1} \end{pmatrix} P,$$

where P is an orthogonal $N \times N$ matrix. We thus have

$$K(\tau\, H) - O(\tau\, H) = P^T\, \mathcal{E}\, P,$$

where $\mathcal{E}$ is the block-diagonal matrix (with 2×2 matrices at the diagonal)

$$\begin{pmatrix} K(\tau E_0) - O(\tau E_0) & 0 & \cdots & 0 \\ 0 & K(\tau E_1) - O(\tau E_1) & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & K(\tau E_{N-1}) - O(\tau E_{N-1}) \end{pmatrix},$$

and therefore

$$\|K(\tau H) - O(\tau H)\| \leq \|\mathcal{E}\| = \max_{j=0,1,\ldots,N-1} \|K(\tau E_j) - O(\tau E_j)\|.$$

Since $|E_j| \leq \beta$, $j = 0, 1, \ldots, N-1$, we finally arrive at

$$\|(q^* + i\, p^*) - \mathrm{e}^{-i\,\tau\, H}(q + i\, p)\| \leq \epsilon(\beta\, \tau)\, \|q + i\, p\|, \tag{28}$$

where

$$\epsilon(\theta) = \sup_{-\theta \leq y \leq \theta} \|K(y) - O(y)\|. \tag{29}$$

By taking into account that $\det K(y) \equiv 1$, the 2-norm of the 2×2 matrix $K(y) - O(y)$ can be explicitly computed to give

$$\|K(y) - O(y)\| = \sqrt{(C(y) - \cos(y))^2 + (S(y) - \sin(y))^2} + \sqrt{C(y)^2 + S(y)^2 - 1},$$

where

$$C(y) = \frac{1}{2}(K_{11}(y) + K_{22}(y)), \quad S(y) = \frac{1}{2}(K_{12}(y) - K_{21}(y)). \tag{30}$$

Notice that $\det K(y) \equiv 1$ implies

$$C(y)^2 + S(y)^2 - 1 = \frac{1}{4}(K_{11}(y) - K_{22}(y))^2 + \frac{1}{4}(K_{12}(y) + K_{21}(y))^2$$

and thus $C(y)^2 + S(y)^2 - 1 \geq 0$ for all real values of y.

3.2.2. Error estimates for several steps of a splitting method

Ideally, given a positive integer m and $\theta = \beta\,\tau > 0$, one would like to determine a sequence (26) of real numbers so that $\epsilon(\theta)$ is minimized. The error bound $\epsilon(\theta)$ being small implies that the $(2m)$th degree polynomial $C(y)$ (resp. the $(2m+1)$th degree polynomial $S(y)$) is a good polynomial approximation of $\cos(y)$ (resp. $\sin(y)$) for $y \in [-\theta, \theta]$, which implies that increasingly large values of $\theta = \beta\,\tau$ will require longer sequences of coefficients (that is, larger values of m), and consequently more computational work. The situation here is in complete analogy with what happened to Taylor and Chebyshev polynomial approximations in the previous section.

By applying the methodology exposed in [3] we have determined several sequences (26) of length $2m+1$ of (near-to-optimal) coefficients for m up to 60. The procedure is described in detail in Appendix A. As shown there, the task is by no means trivial, and severe technical difficulties arise when trying to extend the procedure to arbitrarily large values of $\theta = \beta\,\tau$ (and hence arbitrarily long sequences of coefficients). This is in contrast with Taylor and Chebyshev approximations.

This drawback can always be circumvented by approximating the solution $z(\tau) = O(\tau\, H)z(0)$ of the system of ordinary differential equations (19) in the standard step-by-step way. In our case, approximating $z(\tau)$ in n substeps of length

$$\Delta\tau = \frac{\tau}{n}$$

simply consists in approximating $O(\tau\, H)z(0) = O(n\,\Delta\tau\, H)z(0) = O(\Delta\tau\, H)^n z(0)$ by the vector $K(\Delta\tau\, H)^n z(0)$, where $K(y)$ is a 2×2 matrix with polynomial entries (defined in terms of the sequence (26) as before) that should approximate the rotation matrix $O(y)$ for $y \in [-\frac{\beta\,\tau}{n}, \frac{\beta\,\tau}{n}]$.

Clearly, the resulting procedure for approximating $\mathrm{e}^{-i\,\tau\, H} v$ can be written as an algorithm of the form (25), corresponding to a sequence of coefficients (with a $(2m)$-periodic pattern) of length $2nm+1$. The corresponding error can be estimated as

$$\begin{aligned} \|(q^* + i\, p^*) - \mathrm{e}^{-i\,\tau\, H}(q + i\, p)\| &\leq \|K(\Delta\tau H)^n - O(n\Delta\tau H)\|\, \|q + i\, p\| \\ &\leq \epsilon^{(n)}(\beta\,\Delta\tau)\, \|q + i\, p\|, \end{aligned} \tag{31}$$

where

$$\epsilon^{(n)}(\theta) = \sup_{-\theta \leq y \leq \theta} \|K(y)^n - O(n\, y)\|.$$

Table 1
Relevant parameters of several symplectic splitting methods especially designed to approximate $e^{-i\tau H}v$ with one or more substeps of scaled length $\theta/m = \beta\,\Delta\tau/m$. Here y_* stands for the stability threshold and $\epsilon(\theta)$, $\mu(\theta)$, $\nu(\theta)$, and $\delta(\theta)$ (for $\theta = \beta\Delta\tau_{\max}$) are the coefficients (appearing in the error estimates obtained in Subsection 3.2) given in (29), (32), and (35) respectively.

$M_m^{(\theta/m)}$	m	$\theta = \beta\Delta\tau_{\max}$	y_*/m	$\epsilon(\theta)$	$\mu(\theta)$	$\nu(\theta)$	$\delta(\theta)$
$M_{10}^{(0.5)}$	10	5	0.63	3.6×10^{-8}	8.7×10^{-11}	9.8×10^{-8}	3.6×10^{-8}
$M_{10}^{(0.9)}$	10	9	0.94	3.4×10^{-5}	2.9×10^{-5}	1.1×10^{-5}	6.0×10^{-6}
$M_{20}^{(0.6)}$	20	12	0.79	1.6×10^{-13}	1.4×10^{-13}	5.8×10^{-14}	2.5×10^{-14}
$M_{20}^{(1)}$	20	20	1.1	4.1×10^{-7}	1.8×10^{-8}	4.8×10^{-7}	4.0×10^{-7}
$M_{30}^{(0.75)}$	30	22.5	0.84	8.1×10^{-15}	3.3×10^{-16}	1.5×10^{-14}	7.9×10^{-15}
$M_{30}^{(1)}$	30	30	1.0	4.1×10^{-10}	1.9×10^{-10}	3.1×10^{-10}	2.6×10^{-10}
$M_{30}^{(1.3)}$	30	39	1.36	2.3×10^{-5}	5.2×10^{-6}	2.2×10^{-5}	2.0×10^{-5}
$M_{40}^{(1)}$	40	40	1.1	1.8×10^{-12}	4.9×10^{-14}	2.4×10^{-12}	1.8×10^{-12}
$M_{40}^{(1.2)}$	40	48	1.26	2.1×10^{-8}	2.1×10^{-8}	5.3×10^{-10}	4.7×10^{-10}
$M_{40}^{(1.4)}$	40	56	1.48	1.48×10^{-5}	4.0×10^{-6}	1.7×10^{-5}	1.7×10^{-5}
$M_{50}^{(1)}$	50	50	1.07	4.5×10^{-15}	4.5×10^{-15}	2.0×10^{-17}	1.8×10^{-17}
$M_{50}^{(1.1)}$	50	55	1.13	4.5×10^{-13}	4.2×10^{-13}	4.1×10^{-14}	3.5×10^{-14}
$M_{50}^{(1.2)}$	50	60	1.26	5.4×10^{-11}	2.7×10^{-11}	3.8×10^{-11}	3.4×10^{-11}
$M_{50}^{(1.3)a}$	50	65	1.32	1.2×10^{-8}	1.2×10^{-8}	8.3×10^{-10}	7.6×10^{-10}
$M_{50}^{(1.3)b}$	50	65	1.32	5.9×10^{-7}	9.5×10^{-11}	6.1×10^{-7}	5.9×10^{-7}
$M_{60}^{(1.1)}$	60	66	1.15	7.2×10^{-15}	7.2×10^{-15}	2.6×10^{-17}	2.2×10^{-17}
$M_{60}^{(1.2)a}$	60	72	1.3	1.5×10^{-12}	1.1×10^{-12}	8.3×10^{-13}	7.5×10^{-13}
$M_{60}^{(1.2)b}$	60	72	1.26	4.2×10^{-11}	6.5×10^{-14}	4.6×10^{-11}	4.2×10^{-11}
$M_{60}^{(1.3)}$	60	78	1.36	1.2×10^{-9}	7.8×10^{-11}	1.2×10^{-9}	1.2×10^{-9}
$M_{60}^{(1.4)a}$	60	84	1.41	8.4×10^{-8}	2.4×10^{-8}	7.4×10^{-8}	7.1×10^{-8}
$M_{60}^{(1.4)b}$	60	84	1.46	2.9×10^{-6}	3.7×10^{-9}	2.9×10^{-6}	2.9×10^{-6}
Strang	1	1	2	1.8×10^{-1}	4.7×10^{-2}	1.5×10^{-1}	1.3×10^{-1}
Strang	1	1.4	2	5.1×10^{-1}	1.5×10^{-1}	4.0×10^{-1}	4.0×10^{-1}
Strang	1	1.9	2	1.34862	0.606472	2.4894	1.1746

Our goal is then to minimize $\epsilon^{(n)}(\theta)$. A reasonable requirement is that $K(y)^n$ be bounded for all n. This only happens in general for a certain range of values of y. One thus defines the stability threshold y_* as the largest non negative real number such that $K(y)^n$ is bounded independently of $n \geq 1$ for all $y \in (-y_*, y_*)$ [2]. In particular, for the sequence (23) corresponding to the application of m steps of the Strang splitting, the stability threshold is $y_* = 2m$. As a matter of fact, $2m$ is precisely the maximal stability threshold a sequence of coefficients (26) of length $2m+1$ can achieve [13].

From the analysis carried out in [3], it is possible to show that

$$\|K(y)^n - O(n\,y)\| \leq 2|\sin(n(\arccos(C(y)) - y)/2)| + \sqrt{\frac{S(y)^2}{1-C(y)^2} - 1} + \frac{1}{2}\left(\frac{S(y)^2}{1-C(y)^2} - 1\right),$$

provided that $y \in [-y_*, y_*]$. This implies that, if $\beta\,\Delta\tau \leq y_*$, then

$$\begin{aligned} \|K(\Delta\tau H)^n - O(n\Delta\tau H)\| &\leq \sup_{-\beta\,\Delta\tau \leq y \leq \beta\,\Delta\tau} \|K(y)^n - O(n\,y)\| = \epsilon^{(n)}(\beta\,\Delta\tau) \\ &\leq n\mu(\beta\,\Delta\tau) + \nu(\beta\,\Delta\tau), \end{aligned} \tag{32}$$

where

$$\mu(\theta) = \sup_{-\theta \leq y \leq \theta} |\arccos(C(y)) - y|, \tag{33}$$

$$\nu(\theta) = \sup_{-\theta \leq y \leq \theta} \sqrt{\frac{S(y)^2}{1-C(y)^2} - 1} + \frac{1}{2}\left(\frac{S(y)^2}{1-C(y)^2} - 1\right). \tag{34}$$

As mentioned before, we have determined several optimized splitting methods of m stages (determined by a sequence of coefficients (26) of length $2m+1$) for m up to 60. The relevant parameters of such splitting methods are collected in Table 1. In this table, $M_m^{(\gamma)}$ refers to a method of m stages, with error coefficients $\epsilon(\theta)$, $\mu(\theta)$, $\nu(\theta)$ optimized for $\theta = \gamma m$. For instance, method $M_{60}^{(1.3)}$ can be used to approximate $e^{-i\tau H}v$ with an error bounded (according to (28) and Table 1) by $1.2 \times 10^{-9}\|v\|$ provided that $\tau \leq 78/\beta$. Furthermore, $\mathrm{e}^{-i\tau H}v$ can be approximated by applying n substeps of length $\Delta\tau = \tau/n \leq \Delta\tau_{\max} := 78/\beta$ of method $M_{60}^{(1.3)}$ with an error bounded (according to (32)) by

$$(7.8n \times 10^{-11} + 1.2 \times 10^{-9})\|v\|.$$

In some cases two methods with the same values of m and $\gamma = \theta/m$ have been collected, in which case they are labeled a and b. For instance, methods $M_{60}^{(1.4)a}$ and $M_{60}^{(1.4)b}$ are both designed to approximate $\mathrm{e}^{-i\tau H}v$ with n substeps of length $\Delta\tau = \tau/n \leq \Delta\tau_{\max} := 84/\beta$ of the method. However, they differ in the actual error estimate (32): in the first case, the error is bounded (provided that $\beta|\tau| \leq 84n$) by $(2.4n \times 10^{-8} + 7.4 \times 10^{-8})\|v\|$, while the second one admits the error estimate $(3.7n \times 10^{-9} + 2.6 \times 10^{-6})\|v\|$. This means that method $M_{60}^{(1.4)a}$ will be more efficient if $\beta|\tau| \leq 10\,452$, and the opposite otherwise.

Thus, given the upper bound β of the spectral radius of H and the target time-step τ, if one wants to approximate $\mathrm{e}^{-i\tau H}v$ by applying n substeps of method $M_{60}^{(1.4)a}$, one should choose the smallest positive integer n such that

$$\frac{\tau}{n} \leq \Delta\tau_{\max} := \frac{84}{\beta}, \quad \text{that is,} \quad n = \mathtt{Ceiling}[\tau\beta/84].$$

For instance, suppose the target time-step τ and the bound β are such that $\tau\beta = 1000$. Then, clearly, $n = 12$, so that 12 substeps of scheme $M_{60}^{(1.4)a}$ have to be applied with $\Delta\tau = 1000/(12\beta) \simeq 83.3/\beta$ to achieve the target time-step τ. In this way one gets an approximation with estimated error of size $3.62 \times 10^{-7}\,\|v\|$ with a computational work ($2nm = 2 \times 12 \times 60 = 1440$ real matrix–vector products of the form Hv) comparable to the use of a Chebyshev polynomial approximation of degree 720. In contrast, to guarantee a similar precision with Chebyshev, a polynomial of degree at least 1135 is required, since this is the minimum value of m such that $\epsilon_m^C(1000)\,\|v\| \leq 3.62 \times 10^{-7}\,\|v\|$, with $\epsilon_m^C(\theta)$ given in (16).

It is worth remarking the error coefficients for Strang splitting method (23) with the same value of $\gamma = \theta/m = 1.4$ (also collected in Table 1) are much larger than for methods $M_{60}^{(1.4)a}$ and $M_{60}^{(1.4)b}$.

3.2.3. *Error estimates for combined splitting methods*

Sometimes it is just more efficient to apply a combination of two different methods instead of n substeps of the same scheme. For instance, suppose that $\tau\beta = 177$ and we have an error tolerance of $\mathtt{tol} = 10^{-7}$. If we use $M_{60}^{(1.4)a}$ then $\tau\beta/84 \simeq 2.1$ so that, according with the previous considerations, method $M_{60}^{(1.4)a}$ has to be used with $n = 3$ substeps of size $\Delta\tau = 59/\beta$, much smaller than the value $\Delta\tau_{\max} = 84/\beta$ for which the scheme has been designed. This would result in an approximation fulfilling the required error tolerance obtained with 360 real matrix–vector products of the form Hv. A better strategy would be the following: apply two substeps of scheme $M_{60}^{(1.4)a}$ with $\Delta\tau_{\max} = 84/\beta$ to approximate $w = \mathrm{e}^{-i\,2\Delta\tau_{\max} H}v$ and then approximating $\mathrm{e}^{-i\,(\tau - 2\Delta\tau_{\max})\,H}w$ by using some other method with less stages. More generally, we take $n = \mathtt{Floor}[\tau\beta/84]$ substeps of length $\Delta\tau_{\max} = 84/\beta$ to get $w = \mathrm{e}^{-i\,n\Delta\tau_{\max} H}v$ and then we approximate $\mathrm{e}^{-i\,(\tau - n\Delta\tau_{\max})\,H}w$ with another method of Table 1 involving less stages.

To decide which method has to be used for this last substep, we need an error estimate for the approximation obtained with such a combination of two methods. Assume that we apply n substeps of length $\widehat{\Delta\tau}$ of a method characterized by a 2×2 matrix $\hat{K}(y)$ with polynomial entries, followed by a step of length $\Delta\tau$ of a method characterized by the matrix $K(y)$, where $\tau = n\widehat{\Delta\tau} + \Delta\tau$. From the preceding considerations, it is enough to estimate $\|K(\Delta\tau\, H)\hat{K}(\widehat{\Delta\tau}\, H)^n - O(\tau\, H)\|$. This can be done in terms of the functions $\hat{\mu}(\theta)$, $\hat{\nu}(\theta)$ associated to $\hat{K}(y)$ as defined in Subsection 3.2.2, and the error function $\epsilon(\theta)$ associated to $K(\theta)$ as in Subsection 3.2.1, together with the following function associated to $K(y)$:

$$\delta(\theta) = \sup_{-\theta \leq y \leq \theta} \|K(y)\| - 1. \tag{35}$$

Indeed, one obtains the following error estimate:

$$\begin{aligned}
\|K(\Delta\tau\, H)\hat{K}(\widehat{\Delta\tau}\, H)^n - O(\tau\, H)\| &\leq \|K(\Delta\tau\, H) - \mathrm{e}^{-i\,\Delta\tau H}\|\,\|O(n\widehat{\Delta\tau} H)\| \\
&\quad + \|K(\Delta\tau\, H)\|\,\|\hat{K}(\widehat{\Delta\tau}\, H)^n - O(n\widehat{\Delta\tau} H)\| \\
&\leq \epsilon(\Delta\tau\beta) + (1 + \delta(\Delta\tau\beta))(n\,\hat{\mu}(\widehat{\Delta\tau}\,\beta) + \hat{\nu}(\widehat{\Delta\tau}\,\beta)).
\end{aligned}$$

Since, as it can be noticed in Table 1, $\delta(\theta) \simeq \epsilon(\theta) \ll 1$, then we can take simply

$$\|K(\Delta\tau\, H)\hat{K}(\widehat{\Delta\tau}\, H)^n - O(\tau\, H)\| \lesssim \epsilon(\Delta\tau\,\beta) + n\,\hat{\mu}(\widehat{\Delta\tau}\,\beta) + \hat{\nu}(\widehat{\Delta\tau}\,\beta). \tag{36}$$

It is worth remarking that such an approximation will require $2(n\hat{m} + m) + 1$ real matrix–vector products of the form Hv, and thus is equivalent in complexity to the application of a (Chebyshev or Taylor) polynomial approximation of degree $n\hat{m} + m$.

4. The final algorithm

Once a set of symplectic splitting methods constructed for providing approximations under different conditions are available (methods collected in Table 1) we still have to design a strategy to select the most appropriate scheme or combination of schemes to approximate $e^{-i\tau H}v$ with the desired accuracy and a as small as possible computational cost.

The user has to provide the values for $E_{\min}$ and $E_{\max}$, a subprogram to compute the product Hv for a given real vector v, the target time-step τ and the desired error tolerance $\texttt{tol}$. The procedure then implements the shifting (7), computes the value of β and determines the normalized Hamiltonian H.

Next, the algorithm determines the most efficient method (or composition of methods) among the list of available schemes which provides the desired result: it chooses the cheapest method with error bounds below such tolerance and, if several methods with the same computational cost (same value of m) satisfy this condition, the algorithm chooses the scheme with the smallest error bound. This can be achieved if one starts the search from the methods with the smallest value of m and, for each value of m, proceeds by decreasing accuracy, i.e. by increasing the value of $\theta = \beta\Delta\tau_{\max}$. For a given value of $\beta\tau$ and $\texttt{tol}$ the algorithm checks for each method if $\beta\tau \le \theta$ and, if this condition is satisfied, then it examines if $\epsilon(\theta) < \texttt{tol}$. This procedure corresponds to the sequence of methods collected in Table 1 from top to bottom.

When applying our algorithm in the context of the time integration of the Schrödinger equation with Hamiltonian operators that changes (slowly) with time, $\beta\tau$ will be typically relatively small, and some of the methods from the table will satisfy both conditions for $\beta\tau$ and $\texttt{tol}$. However, for other applications (in particular when solving the Schrödinger equation with time-independent Hamiltonian operators), it may happen that none of the methods from the table satisfy both conditions for $\beta\tau$ and $\texttt{tol}$. Then the time integration is split, i.e. $\beta\tau$ is divided and a composition of one or several methods is used instead.

Due to the high performance of the methods with the largest number of stages (in this case 60) the algorithm examines the cost of n steps for the six 60-stage methods where $n = \texttt{Floor}[\beta\tau/\Delta\tau_{\max}\beta]$ and the last step is carried using one method from the list of methods. It chooses the cheapest methods with the smaller error bound among the composition of methods which provide the desired accuracy.

In this way, if we denote by $K_m^{(\gamma)}$ the matrix associated to method $M_m^{(\gamma)}$, then the resulting splitting method corresponds to the composition

$$K_m^{(\gamma_2)}(\Delta\tau\beta)\left(\hat{K}_{60}^{(\gamma_1)}(\widehat{\Delta\tau}\beta)\right)^{n_1}, \tag{37}$$

where the algorithm chooses the methods (labeled by γ_1, γ_2, m), the time steps, $\Delta\tau$, $\widehat{\Delta\tau}$, and the value of n_1, where $n_1 = 0$ if the method uses just one step. If $n_1 > 0$ the error bound is given by (36) while for $n_1 = 0$ the error bound is just $\epsilon(\tau\beta)$.

This strategy has been implemented as a Fortran code which is freely available for download at the website [28], together with some notes and examples illustrating the whole procedure.

In order to compare the efficiency of the resulting algorithm with the polynomial approximations based in Taylor and Chebyshev with the error estimates collected in Table 1 we have represented in Fig. 1 the computational work (equivalent to a polynomial approximation of degree m) required for different tolerances and values of $\beta\tau$. Diamonds, squares and circles correspond to the error tolerances 10^{-4}, $2\cdot 10^{-7}$ and 10^{-11}, respectively, obtained with one or several steps of schemes in Table 1. Notice that our algorithm based on symplectic splitting methods provide better accuracy with a considerably reduced computational effort.

Some comments are in order here. First, if the present algorithm to compute $e^{-i\tau H}v$ is used in the context of the time integration of the Schrödinger equation with a time-dependent Hamiltonian, as outlined in the Introduction, the value of $\tau = t_{k+1} - t_k$ needs to be relatively small. Otherwise the Magnus approximation may not be sufficiently accurate. In that case we suggest to choose τ in such a way that $\beta\tau \le 84$ (or 78 or 72, depending on the required accuracy in the approximation) so that the algorithm uses only one step for its computation. For instance, if $\tau = 160/\beta$ and $\texttt{tol} = 10^{-6}$, then the algorithm will use two substeps of length $\Delta\tau = \tau/2$ of the method $M_{60}^{(1.4)a}$ with the same matrix H_k to advance the solution of (3) from time $t = t_{k-1}$ to $t = t_k$. However, the same computational effort and precision for the evaluation of the exponentials will result if one halves the length of the time-step, while the error associated to the Magnus approximation will be smaller in that case (about four times smaller in the case of the first order Magnus approximation). Second, if the space discretization has an increasingly high resolution, then the dimension of the resulting matrix H increases accordingly and $\beta\tau$ may take large values. Our algorithm has also been designed to cover this situation, as the examples in the next section illustrate.

5. Numerical examples

Next we apply the algorithm based on symplectic splitting methods presented in Section 3 to two different examples and compare its main features with Chebyshev and Taylor polynomial approximations. For the first example, previously considered in [19] to illustrate Chebyshev and Lanczos approximations, we provide in addition the codes we have produced to generate the results and figures collected here. These can be found at [28]. The second example illustrates the performance of the methods on a one-dimensional Schrödinger equation with a smooth potential.

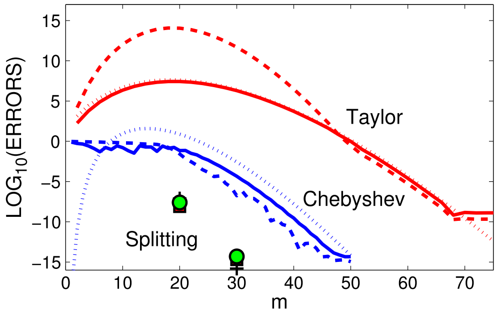

Fig. 2. Different approximations to $e^{-i\tau\beta H}v$, with H given in (38)–(39), v a random vector, $\beta=1$ and $\tau\beta=20$ versus the degree of the polynomials, m. The figure shows the relative error in energy (dashed lines), the error in unitarity (solid lines) and error bounds (dotted lines) for Chebyshev and Taylor methods. The results for the first two splitting methods with $\beta\tau_{\max}\geq\beta\tau=20$, $M_{20}^{(1)}$ and $M_{30}^{(0.75)}$, are also shown: relative error in energy (filled squares), error in unitarity (filled squares) and error bounds (crosses).

Example 1. The problem consists in computing $u(\tau)=\exp(-i\tau\widetilde{H})v$ with $v\in\mathbb{C}^N$ a unitary random vector and the tridiagonal matrix

$$\widetilde{H}=\frac{1}{2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}\in\mathbb{R}^{N\times N}. \tag{38}$$

The eigenvalues of $\widetilde{H}$ verify $0\leq E_k\leq 2$ for all k, so that we can take $E_{\min}=0$, $E_{\max}=2$, and thus $\alpha=\beta=1$ in (7). In consequence, the problem reduces to approximate

$$e^{-i\alpha\tau}e^{-i\beta\tau H}v,\qquad\text{where}\qquad H=\widetilde{H}-I. \tag{39}$$

We take $N=10\,000$ for the numerical experiments, but the results are largely independent of N (this is so even for the simplest, scalar case $N=1$).

Both Chebyshev and Taylor methods have been implemented in such a way that only real valued matrix–vector products are used (we always separate into the real and imaginary parts, i.e. $Hu=H(q+ip)=Hq+iHp$), so that Chebyshev requires to store only 7 real vectors instead of 4 complex vectors.

We take as final time $\tau=20$ and measure the error in energy, the error in the preservation of unitarity and the tolerance for different values of m, the degree of the corresponding polynomials. The results are shown in Fig. 2 with the following notation: dashed lines for the relative error in energy, solid lines for the error in unitarity, and dotted lines for the theoretical error bounds of the approximate solutions.

From the figure it is clear that the theoretical error bounds for the Taylor method are quite accurate for this example (since the bounds for $E_{\min}$ and $E_{\max}$ are sharp) and that for the effective time-step $\tau\beta$ considered, the error is exceedingly large for m below reaching the super linear convergence regime. This is not the case for the Chebyshev method (notice that the estimate (16) is valid only for $m>\tau\beta$) since the coefficients c_k of the polynomial (17) do not grow as much as in Taylor. We also depict the results achieved by the first two splitting methods with $\tau_{\max}\beta\geq\tau\beta=20$, $M_{20}^{(1)}$ and $M_{30}^{(0.75)}$. For these schemes the corresponding relative error in energy is represented by filled squares, the error in unitarity by filled circles and the error bounds by crosses.

The relative performance of different numerical integrators is usually tested by measuring the error of the methods versus their computational cost. However, the splitting methods we are considering in this work are designed to achieve a given tolerance, whereas their computational cost is determined through the error bound estimate. For this reason, we believe it is more appropriate to measure the cost of the methods for different values of the tolerance. In particular, we take $\texttt{tol}=10^{-k}$, $k=1,2,\ldots,12$ and final integration times $\tau=20,50,100,200,500,1000$. Fig. 3 shows the results obtained with Chebyshev (line with squares) and the algorithm based on splitting schemes (line with circles) as a function of m. Even when high accuracy is required over long integration times (the most advantageous situation for Chebyshev approximations), the new algorithm requires a smaller value of m and therefore less computational effort. Notice how the algorithm selects the value of m to achieve the desired tolerance.

Fig. 4 shows the corresponding results for the relative error in energy versus m for the same example. Similar results are obtained for the error in unitarity or the two-norm error for which the error bounds apply (in this case one should compute numerically the exact solution and compare with the approximations obtained for each value of $\texttt{tol}$).

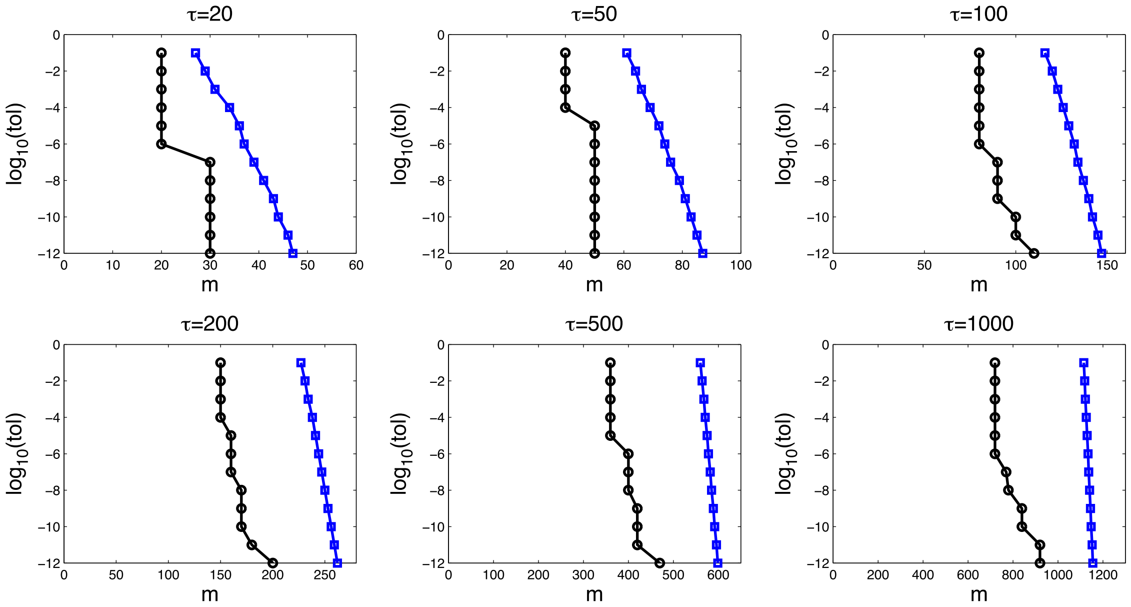

Fig. 3. Degree m of the polynomials to achieve tolerances $\mathtt{tol} = 10^{-k}$, $k = 1, 2, \ldots, 12$ for different values of $\beta\tau$ ($\beta = 1$ for this problem) as determined by the error bound formulas using the Chebyshev method (squares) and the algorithm based on splitting methods (circles).

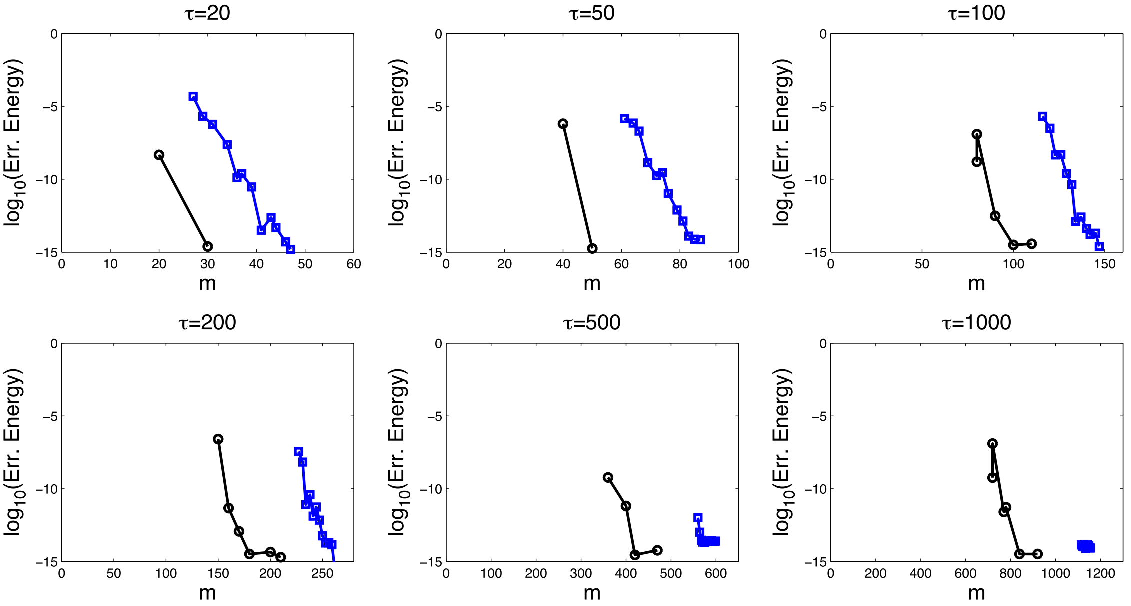

Fig. 4. Same as Fig. 3 but replacing the value of the tolerance tol by the relative error in energy.

Example 2 *(Pöschl–Teller potential).* To illustrate how the methods work on a more realistic case, we consider the well known one-dimensional Pöschl–Teller potential, which is an anharmonic quantum potential

$$V(x) = -\frac{a^2}{2\mu}\frac{\lambda(\lambda-1)}{\cosh^2(ax)},$$

with $a > 0$, $\lambda > 1$. It has been frequently used in polyatomic molecular simulation and is also of interest in supersymmetry, group symmetry, the study of solitons, etc. [6,8,18]. The parameter λ gives the depth of the well, whereas a is related to the range of the potential. The energies are

$$E_k = -\frac{a^2}{2\mu}(\lambda - 1 - k)^2, \qquad \text{with } 0 \le k \le \lambda - 1.$$

Table 2
Bounds to the spectral radius and shifting for the Pöschl–Teller potential with the parameters considered in the text, when the space interval $x \in [-5, 5]$ is split into N parts.

N	E_{min}	E_{max}	α	β
64	−0.65988	0.11583	−0.27202	0.38785
128	−0.65988	0.46333	−0.098275	0.5616
256	−0.65988	1.8533	0.59672	1.2566
512	−0.65988	7.4133	3.3767	4.0366
1024	−0.65988	29.653	14.496	15.156

Table 3
Number of matrix–vector products (in bold) and actual errors given by the Taylor, Chebyshev and our algorithm for different $\tau\beta$ and tolerances `tol`.

	Taylor	Chebyshev	Symplectic
$\tau\beta = 26.4648$	**104**	**51**	**30**
$\texttt{tol} = 10^{-9}$	3.4×10^{-12}	3.7×10^{-12}	4.2×10^{-11}
$\tau\beta = 507.254$	**1836**	**587**	**370**
$\texttt{tol} = 10^{-6}$	2.5×10^{-8}	3.4×10^{-15}	4.4×10^{-9}

We take the following values for the parameters (in atomic units, a.u.): reduced mass $\mu = 1745$ a.u., $a = 2$, $\lambda = 24.5$ (leading to 24 bounded states), and $x \in [-5, 5]$. Moreover, to apply a pseudo spectral space discretization we assume periodicity of the potential in this range. The resulting $V(x)$ is thus continuous and very close to differentiable for all $x \in \mathbb{R}$. Table 2 collects the bounds to the spectral radius (obtained according to (6)) and the corresponding shifting for the Pöschl–Teller potential when the space interval $x \in [-5, 5]$ is split into N parts and for different values of N. Notice how sensibly E_{max} depends on the space discretization.

We take as initial condition a Gaussian function, $\psi(x, 0) = \sigma\, \mathrm{e}^{-(3x)^2}$, where σ is a normalizing constant, so the function and all its derivatives of practical interest vanish up to round off accuracy at the boundaries. The initial conditions contain part of the continuous spectrum, but this fact is largely irrelevant due to the smoothness of the periodic potential and wave function.

Suppose that one is interested in solving the corresponding semi discretized problem in time with the following requirements:

(I) $N = 128$, $\tau = 15\pi$, $\texttt{tol} = 10^{-9}$. In this case $\tau\beta = 26.4648$.
(II) $N = 512$, $\tau = 40\pi$, $\texttt{tol} = 10^{-6}$. Now $\tau\beta = 507.254$.

We have to determine first, of course, the degree m of the polynomial from the corresponding error bounds (for Taylor the time interval is divided by two in (I) and by 36 in (II) to avoid exceedingly large round off errors). Table 3 shows the number of matrix–vector products used by each method (in bold) and the 2-norm error for each method (compared with the exact solution obtained numerically with very high accuracy). In the first case our algorithm makes the computations in a single step using $M_{30}^{(1)}$ while in the second case it uses 6 steps of the scheme $M_{60}^{(1.4)a}$ followed by one step of $M_{10}^{(0.5)}$, i.e. the composition (37) is now

$$K_{10}^{(0.5)}(\Delta\tau\beta)\left(\hat{K}_{60}^{(1.4)a}(\widehat{\Delta\tau}\beta)\right)^{6}$$

with $\widehat{\Delta\tau} = 84/\beta$ and $\Delta\tau = 40\pi - 6\widehat{\Delta\tau}$, and for a total of 370 products. Again, the algorithm based on symplectic splitting methods is able to produce results with the required accuracy with less computational effort.

Acknowledgements

The authors acknowledge Ministerio de Economía y Competitividad (Spain) for financial support through the coordinated project MTM2013-46553-C3. AM is additionally partially supported by the Basque Government (Consolidated Research Group IT649-13), and FC by NPRP GRANT #5-674-1-114 from the Qatar National Research Fund. We are grateful to the anonymous referees, whose comments have helped us to improve this paper.

Appendix A. Construction of methods

We next describe the algorithm used to determine the coefficients (26) of length $2m + 1$ for given m and $\theta \in (0, 2m)$.

Since all the error estimates in Subsection 3.2 depend exclusively on the even polynomial (of degree $2m$) $C(y)$ and the odd polynomial (of degree $2m + 1$) $S(y)$ given in (30), we first try to determine an appropriate pair of such polynomials satisfying the necessary conditions $C(0) = 1$ and $C(x)^2 + S(x)^2 - 1 > 0$ (for all $x \in \mathbb{R}$). Such pair of polynomials is uniquely determined by a polynomial $P(y) = C(y) + S(y)$ of degree $2m+1$ satisfying

$$P(0) = 1, \qquad \frac{1}{2}(P(y)^2 + P(-y)^2) - 1 \geq 0. \tag{40}$$

Once an appropriate polynomial $P(y) = C(y) + S(y)$ satisfying (40) is chosen, there is only a finite number of corresponding sequences (26), which can be effectively determined [2]. Since all of them share the same error estimates, we choose among them a sequence that minimizes

$$\sum_{j=1}^{m+1} |a_j| + \sum_{j=1}^{m} |b_j|.$$

We next focus on the effective construction of the polynomial $P(y) = C(y) + S(y)$ of degree $2m+1$.

On the one hand, in order that the expression $\sqrt{C(y)^2 + S(y)^2 - 1}$ featuring in the error estimate (29) be small in the interval $y \in [-\theta, \theta]$,

$$\sup_{-\theta \leq y \leq \theta} |\cos(y + e(y)) + \sin(y + e(y)) - P(y)| \tag{41}$$

should be small for some real valued function $e(y)$. On the other hand, minimizing

$$\sqrt{(C(y) - \cos(y))^2 + (S(y) - \sin(y))^2}$$

in the interval $y \in [-\theta, \theta]$ is, provided that (41) is small enough, essentially equivalent to minimizing

$$\sup_{-\theta \leq y \leq \theta} |e(y)|. \tag{42}$$

To reduce the complexity of the final algorithm for determining the polynomial $P(y)$, we will try to minimize instead an alternative norm of $e(y)$ that we introduce next. First observe that if

$$e(y) = \hat{e}_0 + \sum_{j \geq 1} \hat{e}_j T_j(y/\theta) \tag{43}$$

is the Chebyshev series expansion of the function $e(y)$, then

$$\sup_{-\theta \leq y \leq \theta} |e(y)| \leq \sum_{j \geq 0} |\hat{e}_j|. \tag{44}$$

This suggests that the right-hand side of (44) may be a good alternative to the supremum norm for sufficiently smooth functions $e(y)$. For practical considerations, we will minimize instead the following alternative norm of the function $e(y)$

$$\|e\|_\theta \equiv \sqrt{\sum_{j \geq 0} (\hat{e}_j)^2}. \tag{45}$$

Now, to determine the polynomial $P(y) = C(y) + S(y)$ of degree $2m+1$, we consider, for a given odd integer l such that $m+1 \leq l \leq 2m$, a given set of nodes $y_1, \ldots, y_l$ symmetrically placed in the interval $[-\theta, \theta]$, and a given odd polynomial $e(y)$ of degree $l-2$, the polynomial $P(y)$ of degree $2l-1$ interpolating in the Hermite sense the function $\cos(y + e(y)) + \sin(y + e(y))$ for the nodes $y_1, \ldots, y_l$. In particular, this implies that $P(0) = 1$ and

$$C(y)^2 + S(y)^2 - 1 = \frac{1}{2}(P(y)^2 + P(-y)^2) - 1 = V(y) W(y)^2 \tag{46}$$

where $W(y) = (y - y_1) \cdots (y - y_l)$, and $V(y)$ is an even polynomial of degree $4m - 2l + 2$. Thus, $P(y)$ satisfies the necessary condition (40) if and only if $V(y) \geq 0$ for all y.

Notice that the interpolation error (41) admits an upper bound of the form

$$\sup_{-\theta \leq y \leq \theta} |\cos(y + e(y)) + \sin(y + e(y)) - P(y)| \leq \frac{\eta}{(2l)!} \sup_{-\theta \leq y \leq \theta} W(y)^2, \tag{47}$$

where $\eta > 0$ is an upper bound of the (absolute value of) the $(2l)$th derivative of the function $\cos(y + e(y)) + \sin(y + e(y))$ in the interval $y \in [-\theta, \theta]$.

For a prescribed set of nodes $y_1, \ldots, y_l$, we restrict the choice of the odd polynomial $e(y)$ (of degree $l-2$) so that the Hermite interpolating polynomial $P(y)$ is of degree $2m+1$ (which introduces $2(l-m)-2$ non-linear constraints on the non-zero coefficients $\hat{e}_1, \hat{e}_3, \ldots, \hat{e}_l$ of the polynomial $e(y)$ given by (43)), and determine $e(y)$ by minimizing the norm $\|e\|_\theta$

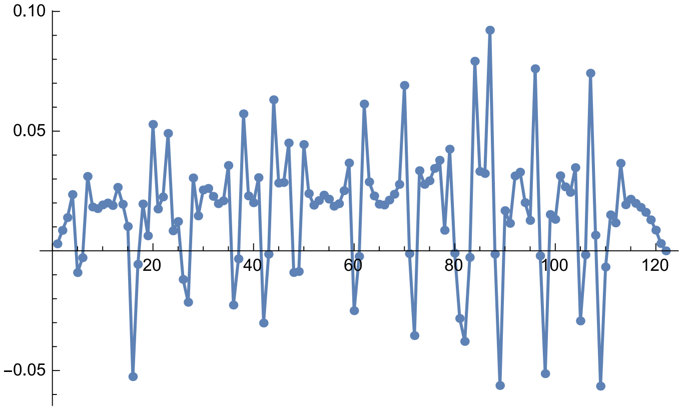

Fig. 5. Graphical representation of sequence $(a_1, b_1, a_2, b_2, \ldots, a_{60}, b_{60}, a_{61})$ of method $M_{60}^{(1.4)a}$ in Table 1, obtained with $\theta = 84$ and $l = 97$.

for that restricted set of odd polynomials $e(y)$ of degree $l - 2$. This produces a polynomial $P(y)$ for each choice of the set of nodes $y_1, \ldots, y_l$. It then remains to choose, for a prescribed positive odd integer l, such a set of nodes $y_1, \ldots, y_l$.

The error estimate (47) suggests that a good choice for the interpolating nodes $\{y_1, \ldots, y_l\}$ may be given by the zeros of the Chebyshev polynomial $T_l(y/\theta)$ of degree l, which corresponds to minimizing the supremum norm (in the interval $[-\theta, \theta]$) of the polynomial $W(y)$. Notice that minimizing the alternative norm $\|W\|_\theta$ also gives rise to the same set of nodes. It then only remains, for given odd positive number $2m + 1$ and for given $\theta > 0$, to determine the number l of interpolating nodes, that should satisfy $m + 1 \le l \le 2m$. If l is too close to $2m$, then, very few degrees of freedom are left to minimize $\|e\|_\theta$, and if l is too close to $m + 1$, then the Hermite interpolating error (41) is too large, causing the norm of the function $C(y)^2 + S(y)^2 - 1$ not being small enough, in addition to $V(y)$ in (46) typically not being positive. We thus proceed by determining $P(y) = C(y) + S(y)$ for different values of l close to $(3m + 3)/2$, and choosing, among those satisfying $V(y) \ge 0$, one having the best error coefficient $\epsilon(\theta)$ defined in (29).

Unfortunately, choosing the interpolating nodes $\{y_1, \ldots, y_l\}$ as the zeros of the Chebyshev polynomial $T_l(y/\theta)$ of degree l typically results in a polynomial $P(y) = C(y) + S(y)$ that does not satisfy the stability condition

$$|C(y)| \le 1, \quad y \in [-\theta, \theta], \tag{48}$$

so that the error coefficients $\mu(\theta), \nu(\theta)$ are not well defined, and thus the resulting splitting method cannot be reliably used in a step-by-step manner for large values of $\beta\tau$. In order to produce splitting methods satisfying that stability condition for given θ, we proceed iteratively to choose the interpolating nodes $\{y_1, \ldots, y_l\}$ and the corresponding polynomial $P(y)$ as follows: As a first approximation, we require the set of nodes $\{y_1, \ldots, y_l\}$ to contain the set $\{j\pi \ : \ j \in \mathbb{Z}, \ |j\pi| \le \theta\}$ and determine the remaining nodes by minimizing the norm $\|W\|_\theta$ of $W(y) = (y - y_1) \cdots (y - y_l)$. Once the polynomial $P(y) = C(y) + S(y)$ is determined for that set of nodes $\{y_1, \ldots, y_l\}$, we compute the set of zeros of $C'(y) = 0$ that are included in the interval $[-\theta, \theta]$ (that are typically close to $\{j\pi \ : \ j \in \mathbb{Z}, \ |j\pi| \le \theta\}$), and determine the remaining nodes by minimizing the norm $\|W\|_\theta$ of $W(y) = (y - y_1) \cdots (y - y_l)$. Successive iteration of this process gives a sequence of polynomials $P(y) = C(y) + S(y)$ that converge to a polynomial satisfying the stability condition (48).

As an example, we have obtained the method $M_{60}^{(1.4)a}$ in Table 1 by following this procedure for $m = 60$, $\theta = 84$, and $l = 97$, which has produced a splitting methods with sequence of coefficients (26) plotted in Fig. 5.

References

[1] S. Blanes, F. Casas, A. Farrés, J. Laskar, J. Makazaga, A. Murua, New families of symplectic splitting methods for numerical integration in dynamical astronomy, Appl. Numer. Math. 68 (2013) 58–72.

[2] S. Blanes, F. Casas, A. Murua, On the linear stability of splitting methods, Found. Comput. Math. 8 (2008) 357–393.

[3] S. Blanes, F. Casas, A. Murua, Error analysis of splitting methods for the time dependent Schrödinger equation, SIAM J. Sci. Comput. 33 (2011) 1525–1548.

[4] S. Blanes, F. Casas, J.A. Oteo, J. Ros, The Magnus expansion and some of its applications, Phys. Rep. 470 (2009) 151–238.

[5] S.A. Chin, C.R. Chen, Gradient symplectic algorithms for solving the Schrödinger equation with time-dependent potentials, J. Chem. Phys. 117 (2002) 1409–1415.

[6] S.-H. Dong, Factorization Method in Quantum Mechanics, Springer, 2007.

[7] M.D. Feit, J.A. Fleck Jr., A. Steiger, Solution of the Schrödinger equation by a spectral method, J. Comput. Phys. 47 (1982) 412–433.

[8] S. Flügge, Practical Quantum Mechanics, Springer, 1971.

[9] S. Gray, D.E. Manolopoulos, Symplectic integrators tailored to the time-dependent Schrödinger equation, J. Chem. Phys. 104 (1996) 7099–7112.

[10] S. Gray, J.M. Verosky, Classical Hamiltonian structures in wave packet dynamics, J. Chem. Phys. 100 (1994) 5011–5022.

[11] T.Z. Huang, R.S. Rau, A simple estimation for the spectral radius of (block) H-matrices, J. Comput. Appl. Math. 177 (2005) 455–459.

[12] T. Jahnke, Ch. Lubich, Error bounds for exponential operator splittings, BIT 40 (4) (2000) 735–744.

[13] R. Jeltsch, O. Nevanlinna, Stability of explicit time discretizations for solving initial value problems, Numer. Math. 37 (1981) 61–91.

[14] D. Kosloff, R. Kosloff, A Fourier method solution for the time dependent Schrödinger equation as a tool in molecular dynamics, J. Comput. Phys. 52 (1983) 35–53.

[15] R. Kosloff, Time-dependent quantum mechanical methods for molecular dynamics, J. Phys. Chem. 92 (1988) 2087–2100.

[16] D. Lauvergnat, S. Blasco, X. Chapuisat, A simple and efficient evolution operator for time-dependent Hamiltonians: the Taylor expansion, J. Chem. Phys. 126 (2007) 204103.

[17] C. Leforestier, R.H. Bisseling, C. Cerjan, M.D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, R. Kosloff, A comparison of different propagation schemes for the time dependent Schrödinger equation, J. Comput. Phys. 94 (1991) 59–80.

[18] R. Lemus, R. Bernal, Connection of the vibron model with the modified Pöschl–Teller potential in configuration, Chem. Phys. 283 (2002) 401–417.

[19] C. Lubich, From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis, European Mathematical Society, 2008.

[20] W. Magnus, On the exponential solution of differential equations for a linear operator, Commun. Pure Appl. Math. VII (1954) 649–673.

[21] G. Mazzi, B.J. Leimkuhler, Dimensional reductions for the computation of time-dependent quantum expectations, SIAM J. Sci. Comput. 33 (2011) 2024–2038.

[22] M. Mulansky, Simulating DNLS models, Technical report, 2013, arXiv:1304.1608.

[23] C. Neuhauser, M. Thalhammer, On the convergence of splitting methods for linear evolutionary Schrödinger equations involving an unbounded potential, BIT 49 (2009) 199–215.

[24] F.W.J. Olver, D.W. Lozier, R.F. Boisvert, C.W. Clark, NIST Handbook of Mathematical Functions, Cambridge University Press, 2010.

[25] T.J. Park, J.C. Light, Unitary quantum time evolution by iterative Lanczos reduction, J. Chem. Phys. 85 (1986) 5870–5876.

[26] Ch. Skokos, E. Gerlach, J.D. Bodyfelt, G. Papamikos, S. Eggl, High order three part split symplectic integrators: efficient techniques for the long time simulation of the disordered discrete nonlinear Schrödinger equation, Phys. Lett. A 378 (2014) 1809–1815.

[27] H. Tal-Ezer, R. Kosloff, An accurate and efficient scheme for propagating the time dependent Schrödinger equation, J. Chem. Phys. 81 (1984) 3967–3971.

[28] An efficient algorithm for the time integration of the Schrödinger equation, http://www.gicas.uji.es/software.html.

[29] M. Thalhammer, High-order exponential operator splitting methods for time-dependent Schrödinger equations, SIAM J. Numer. Anal. 46 (2008) 2022–2038.

[30] M. Thalhammer, Convergence analysis of high-order time-splitting pseudo-spectral methods for nonlinear Schrödinger equations, SIAM J. Numer. Anal. 50 (2012) 3231–3258.

[31] M. Yang, A simple method for estimating the bounds of spectral radius of nonnegative irreducible matrices, Appl. Math. E-Notes 11 (2011) 67–72.

[32] Q. Zhu, G.D. Hu, L. Zeng, Estimating the spectral radius of a real matrix by discrete Lyapunov equation, J. Differ. Equ. Appl. 17 (2011) 603–611.