ELSEVIER

Applied Numerical Mathematics 25 (1997) 275–286

APPLIED NUMERICAL MATHEMATICS

Optimal stability polynomials for splitting methods, with application to the time-dependent Schrödinger equation

Robert I. McLachlan [a,*], Stephen K. Gray [b]

[a] *Isaac Newton Institute for Mathematical Sciences, University of Cambridge, Cambridge CB3 OEH, UK*
[b] *Chemistry Division, Argonne National Laboratory, Argonne, IL 60439, USA*

Abstract

We determine optimal stability polynomials $p(x)$ for splitting method solutions of differential equations, building on previous work by López-Marcos, Sanz-Serna and Skeel (1996). The methods have a variety of stage numbers and are up to eighth order. Knowledge of $p(x)$ allows construction of the most stable splitting methods for given complexity. As an illustration, we construct symplectic corrector algorithms ($C^{-1}KC$, where the kernel K is an m-stage splitting method) which approximate the solution of linear Hamiltonian systems. The kernels K that realize the optimal stability polynomials are found for this case. We also discuss the construction of correctors C, and find them for two particularly promising kernels. Numerical calculations for a time-dependent Schrödinger equation problem confirm the methods' usefulness. © 1997 Elsevier Science B.V.

1. Introduction

Splitting methods for numerically integrating differential equations, while having a long history, are still common and useful tools in many physical applications. There are two good reasons for this. First, many such methods are simple and easy to implement, e.g., the leapfrog (Verlet, Störmer) method which is a staple of large scale molecular dynamics simulations. Second, important mathematical and physical properties of the differential equations can be preserved by splitting methods. In the leapfrog example, the symplectic structure is preserved [10]. Splitting methods are based on splitting the vector field of a differential equation into a sum of two solvable (or at least easier to handle) parts A and B, and then approximating the exact time evolution as a composition of time evolutions of the parts. An m-stage splitting method is of the form

$$M_m(\tau) = M_A(a_m\tau)M_B(b_m\tau)\cdots M_A(a_2\tau)M_B(b_2\tau)M_A(a_1\tau)M_B(b_1\tau), \tag{1}$$

* Corresponding author. Permanent address: Department of Mathematics, Massey University, Palmerston North, New Zealand. E-mail: R.McLachlan@massey.ac.nz.

PII S0168-9274(97)00064-0

where M_A and M_B are the evolution operators for A and B individually. (See [5,8] for surveys of splitting and composition methods.) If $M(\tau)$ is the exact evolution operator for the full problem, i.e., the operator that takes a vector z from time t to $t+\tau$, then the $2m$ coefficients a_k, b_k of the splitting method are chosen in some fashion as to ensure $M_m \approx M$. It is common to choose the coefficients such that $M_m(\tau) = M(\tau) + \mathrm{O}(\tau^{n+1})$, i.e., such that the method is accurate to order n in the time step. However, another important issue is *stability*. Typically, (1) will be unstable for a given A and B for $|\tau| > \tau^*$. Often methods with relatively high orders n, exhibit such small τ^* as to be practically useless. The aims of this paper are to show how the theoretically best possible stability can achieved for splitting methods with a variety of orders and stages applied to a certain class of problems, and to develop some new splitting methods for linear Hamiltonian systems using these ideas.

To test the stability of the integrator (1) one linearizes it around some state and finds the time steps for which all initial conditions remain bounded for all time, i.e., for which all eigenvalues λ of $M_m(\tau)$ have $|\lambda| \leqslant 1$ and if $|\lambda| = 1$, then this eigenvalue has equal algebraic and geometric multiplicities. (In practice we simply demand unit multiplicity.) However, the stability depends on the state about which one linearizes. Simple methods, such as leapfrog, may show "average" stability when applied to a variety of problems. If stability is optimized about a particular state, then this method may show no better (or even worse) stability when applied to a different state. For a given state, the choice of splitting also affects A and B and hence influences the stability.

Here we consider the stability of (1) applied to the harmonic oscillator with standard (q–p) splitting. This leads to optimal stability methods for any problem in which A and B, when linearized about some state, are simultaneously diagonalizable (i.e., their matrices commute). Although our motivation was the development of efficient methods for the Schrödinger equation, given below, the ideas and methods here apply to a wide variety of spatially discretized Hamiltonian partial differential equations (PDEs).

We are not the first to develop "maximal stability" methods in this context: López-Marcos et al. [3] developed an interesting order $n = 4$ method using a similar approach. Our stability polynomial results may thus be viewed as an extension of their approach to arbitrary order n and stage number m. For example, we find explicit "stability polynomials" up to $n = 8$. These polynomials determine the best possible stability that any method of this class can have in principle; once this is known one can search for methods that actually achieve this theoretical limit.

In particular, we develop suitable splitting methods for the discretized form of the time-dependent Schrödinger equation [2]. Such discretized partial differential equations can be of sufficiently large dimension that direct determination of all the eigenvalues and eigenvectors of the linear system is not possible, which then makes numerical solutions of the linear differential equation an important tool for determining physical observables. We study "symplectic corrector methods" [1,4,6,11], $C^{-1}KC$, where the kernels K and correctors C are both of the form of (1). The advantage is that the corrector need only be rarely (or, depending on the physical information desired, never) applied, and does not affect the stability, since it is only a change of variables. Thus we first determine K to have the known optimal stability, and then determine certain correctors C based on accuracy requirements.

To sum up, we

- determine optimal stability polynomials (Section 2): these determine the best possible stability for any splitting method applied to the Hamiltonian $h = T(p) + V(q)$, where the linear parts $T''(0)$ and $V''(0)$ commute;

- determine kernels K that realize this optimal stability for the time-dependent Schrödinger equation (Section 3);
- determine correctors C that give the whole method the desired order (Section 4).

Finally, Section 5 presents some numerical results.

2. Stability polynomials

Consider a Hamiltonian system with Hamiltonian $h = T(p) + V(q)$, which we solve numerically with a splitting method of the form

$$\begin{aligned} p_k &= p_{k-1} - b_k \tau \nabla V(q_{k-1}), \\ q_k &= q_{k-1} + a_k \tau \nabla T(p_k), \end{aligned} \qquad k = 1, \ldots, m. \tag{2}$$

Suppose $q = p = 0$ is a fixed point, and the linear parts $V''(0)$ and $T''(0)$ commute. Then there are coordinates in which they are both diagonal, and in those coordinates the linear part of the Hamiltonian decouples into systems of the form $h = \frac{1}{2}(\mu q^2 + \lambda p^2)$. Assuming $\mu,\ \lambda > 0$ so that the fixed point is elliptic, we can rescale q, p and time to bring the Hamiltonian into the form $h = \frac{1}{2}(q^2 + p^2)$. The new time is $x = t\sqrt{\lambda\mu}$. Thus, to study stability in this case it suffices to consider the harmonic oscillator. Let $z = (q, p)^{\mathrm{T}}$ and write the exact solution as

$$z(x) = M(x)z(0), \quad M(x) = \begin{pmatrix} \cos x & \sin x \\ -\sin x & \cos x \end{pmatrix} \tag{3}$$

and the numerical solution for an m-stage splitting method, from (1), as

$$z(x) = M_m(x)z(0), \quad M_m(x) = \prod_{i=1}^{m} \begin{pmatrix} 1 - a_i b_i x^2 & a_i x \\ -b_i x & 1 \end{pmatrix}, \tag{4}$$

where we now think of x as the scaled time step.

The exact evolution operator $M(x)$ has the following properties:

(M1) M is area preserving, i.e., $\det M = 1$ for all x;

(M2) M is elliptic, i.e., $|\operatorname{tr} M| \leqslant 2$ for all x;

(M3) M has eigenvalues $\mathrm{e}^{\pm \mathrm{i}x}$;

(M4) M is orthogonal, i.e., $M^{\mathrm{T}}M = I$ for all x.

(M1), the symplectic property, is essential in long-time integrations and is ensured by the use of the splitting method. (M2) is also essential, for otherwise $\lim_{n\to\infty} M^n = \infty$ and the solution z blows up. The numerical evolution operator $M_m(x)$ will not satisfy this for all x, but only

$$|\operatorname{tr} M_m(x)| \leqslant 2, \qquad |x| \leqslant x^*, \tag{5}$$

where x^* is the *stability limit*. Practically, time steps τ for a stable numerical solution must be less than $x^*/|H|_{\max}$, where $|H|_{\max}$ is the maximum eigenvalue product $\sqrt{\lambda\mu}$ of the Hamiltonian. (M3) is less important: getting the eigenvalues slightly wrong only introduces a phase error into the solution, which is impossible to avoid in any event. (M4) is also less important: if (M1) and (M2) are satisfied, but not (M4), then the phase portrait of the numerical solution consists of ellipses instead of circles. In the application to the Schrödinger equation (see Section 5), this leads to errors in the norm of the solution.

It is easy to see that $\operatorname{tr} M_m(x)$ is an even polynomial of order $2m$. Defining the stability polynomial as $p(x) = \frac{1}{2}\operatorname{tr} M_m(x)$, then a necessary condition for M_m to be accurate to order n is

$$p(x) = \sum_{j=0}^{n/2}(-1)^j \frac{x^{2j}}{(2j)!} + \sum_{j=n/2+1}^{m} c_j x^{2j}, \tag{6}$$

where for simplicity we have taken n even.

The c_j coefficients in $p(x)$ are dependent on the particular choice of splitting method coefficients a_k, b_k. The best possible choice of splitting method coefficients in terms of stability, as recognized by López-Marcos et al. [3], is the one that yields c_j coefficients such that $p(x)$ has the largest possible associated stability limit x^*. Therefore one approach is to first determine the optimal c_j, and then try to devise splitting method coefficients that lead to these c_j (as well as satisfy the order conditions).

The problem of determining the c_j is a straightforward optimization problem which in principle can be solved completely. (In practice, however, issues of numerical precision and stability can make this task tricky when m becomes large.) To get $|p(x)| \leqslant 1$, one should make $p(x)$ turn as often as possible; but some turning points are fixed by the given Taylor series. Local optima are obtained by using the c_j to make $|p(x_i)| = 1$ exactly at $m - n/2$ turning points x_i, and experimenting with various values of m and n suggests that these are global optima. That is, one first solves

$$p(x_i) = \pm 1, \quad p'(x_i) = 0, \qquad i = 1, \ldots, m - n/2, \tag{7}$$

for c_j and x_i and then finds the stability of this $p(x)$. Eqs. (7) are $2m - n$ equations in $2m - n$ unknowns. They are linear in the c_j, so in practice one solves half of them analytically for the c_j and then searches numerically for solutions of the remaining polynomial system in the x_i.

How many turning points should be fixed by (7)? Since $p(x)$ is even and has a maximum at $x = 0$, it can have at most $m - 1$ turning points in $x > 0$. The worst case is when all of these have to be fixed by (7); but as n increases, the Taylor polynomial approximation to $\cos x$ has more turning points that approximate those of $\cos x$ anyway.

Consider the turning point of $\cos x$ at $x = \pi$. Note that

$$\sum_{j=0}^{n/2}(-1)^j \frac{x^{2j}}{(2j)!} = \cos x - \frac{(-1)^{n/2+1}}{(n+1)!}\int_0^x (x-\xi)^{n+1}\cos\xi \, \mathrm{d}\xi. \tag{8}$$

So when $c_j = 0$, $p(\pi) < -1$ if $n/2$ is odd and $p(\pi) > -1$ if $n/2$ is even. In the latter case, we can hope that the imposition of (7) does not push $p(x)$ below -1 near $x = \pi$, and thus drop this minimum from (7). This makes $n = 4$, 8 better than $n = 2$, 6. (Unfortunately the Taylor polynomial for $n = 6$ does not have a maximum near $x = 2\pi$, so all turning points must be fixed by (7).)

Example. $n = 4$ (*fourth order*). For $m = 2$ we have $p(x) = 1 - x^2/2 + x^4/24$ so $x^* = \sqrt{12} \approx 3.464$. For $m = 3$ we have $p(x) = 1 - x^2/2 + x^4/24 + cx^6$ and we solve

$$p(x_1) = 1, \qquad p'(x_1) = 0, \tag{9}$$

giving

$$c = -1/1152, \qquad x_1 = 2\sqrt{6}$$

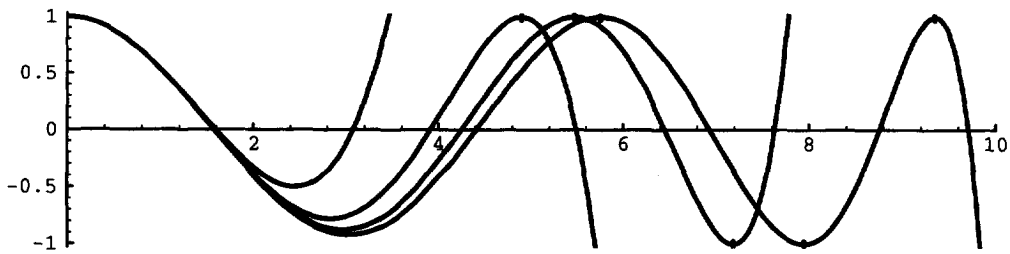

Fig. 1. Optimal stability polynomials for methods of order 4. The optimal polynomials $p(x)$ (see (6)) for $m = 2$, 3, 4 and 5 are shown. Points at which $p(x)$ is constrained to ± 1 are flagged.

Table 1
Optimal stability interval x^* for m-stage, nth order methods

m	n			
	2	4	6	8
2	4	3.464		
3	6	5.695	2.752	
4	8	7.782	4.837	4.635
5	10	9.828	6.762	6.636
6	12		8.663	8.570

and stability limit $x^* = 2(2^{1/3} + 2^{2/3}) = 5.695$. This is the stability polynomial determined by López-Marcos et al. [3]. For $m = 4$, we enforce a maximum at $x_1 \approx 2\pi$, a minimum at $x_2 \approx 3\pi$, and find the stability limit $x^* = 7.782$. This $p(x)$ has a minimum at $p(2.96) = -0.874$, illustrating the argument in the previous paragraph. These polynomials, as well as that for $m = 5$ (which has $x^* = 9.828$), are shown in Fig. 1.

Note that for large x, the methods have poor phase accuracy. In the last example, at $x = 2.96$ the eigenvalues of the numerical evolution operator are $e^{\pm 2.634i}$ instead of the exact $e^{\pm 2.96i}$ from property (M3) (because $\cos^{-1}(-0.874) = 2.634$). Suitable applications for such a method will have small amplitudes in the fast (large x) waves, such as discretizations of PDEs with smooth solutions. The fast waves are only present in the discretization to ensure that the slow waves are accurately represented. We need only ensure that they do not destabilize the whole calculation.

When m is large, solving (7) becomes difficult. We used a mixture of analytic and numerical solution procedures, namely Newton's method and a continuous variable simulated annealing algorithm [9].

Table 1 gives the optimal stability limits x^* for various m and n, and Table 2 gives the stability polynomials themselves. For order $n = 2$, the best one can do is to take m steps of leapfrog. For fixed m, increasing the order n probably always makes the stability worse, as more control is needed at $x = 0$ and we have one turning point less to play with. Conversely, for fixed order n, but increasing m, one can achieve greater stability.

More significantly, consider m/x^*, the last column of Table 2. m/x^* is proportional to the amount of work to integrate for a unit time with maximum stable time step, and represents the relative minimum

Table 2
Various m-stage, nth order stability polynomial coefficients c_j, $j = n/2+1, \ldots, m$, their associated optimal stabilities, x^*, and relative minimum efforts, m/x^*

n	m	c_j	x^*	m/x^*
2	m		$2m$	0.500
4	2	none	3.464	0.577
4	3	$-1/1152$	5.695	0.527
4	4	$-0.110982534333366 \cdot 10^{-2}$	7.782	0.514
		$0.921635062636551 \cdot 10^{-5}$		
4	5	$-0.121407893685527 \cdot 10^{-2}$	9.828	0.509
		$0.144168151282747 \cdot 10^{-4}$		
		$-0.598566579684913 \cdot 10^{-7}$		
6	3	none	2.752	1.090
6	4	$0.2228473740133193 \cdot 10^{-4}$	4.837	0.827
6	5	$0.2421736241649607 \cdot 10^{-4}$	6.762	0.739
		$-0.1968158553850286 \cdot 10^{-6}$		
6	6	$0.246107452341319 \cdot 10^{-4}$	8.663	0.693
		$-0.247651892109466 \cdot 10^{-6}$		
		$0.111265369266638 \cdot 10^{-8}$		
8	4	none	4.635	0.863
8	5	$-0.2140483644066403 \cdot 10^{-6}$	6.636	0.753
8	6	$-0.2556649778998596 \cdot 10^{-6}$	8.570	0.700
		$0.1190911947435882 \cdot 10^{-8}$		

effort of the method. For example, the method with the smallest m/x^* can be run with the least numerical work. The theoretical limit for m/x^* is $\frac{1}{2}$, which is achieved by the leapfrog method, and this is another important reason for the popularity of the leapfrog method. However, leapfrog is only of order $n = 2$, and Table 2 shows that we can find quite reasonable m/x^* values with $n = 4$, 6 and 8. For example, m/x^* is only slightly greater than $\frac{1}{2}$ for the $n = 4$ cases, and we can find $n = 6$ and 8 cases with just a 40% increase in m/x^* to about 0.7. For comparison, note that m/x^* is typically greater than unity for many high order splitting methods (see [2, Table II]). Table 2 also shows that for fixed n, m/x^* *decreases* as m increases. (Indeed, it is interesting to conjecture that as $m \to \infty$, $m/x^* \to \frac{1}{2}$.)

3. Determination of kernels

We now restrict our attention to linear systems, because of our planned application in Section 5. We identify linear maps with the matrices that represent them in the chosen coordinates.

The symplectic corrector algorithm [1,4,6,11] approximates the time evolution of a Hamiltonian system by $C^{-1}KC$, where the kernel K is a symplectic mapping but C, the corrector, may or may not be symplectic. Because $C^{-1}KC$ is similar to K, the eigenvalues and the stability of such a method are determined only by the kernel K. Therefore in generating a suitable algorithm, one might wish to first determine kernels K with the same stability as the stability polynomials discussed above. In fact, the eigenvalues are the *only* invariants of the method, so if we get these correct to order n, there exists a corrector C to give the whole method order n. The minimum number of stages for a given order, namely $n/2$, is much smaller than in the general composition problem. At $n = 8$ we need 4 stages instead of the 15 required in general. This is because we consider only linear Hamiltonians, and use correctors.

We now discuss how to obtain the $2m$ coefficients a_k, b_k if the kernel K is taken to be the m-stage algorithm (2). Recall that $\operatorname{tr} M_m(x)$ is an even polynomial of degree $2m$ with m independent coefficients (its constant term is always 1). These coefficients are functions of the $2m$ parameters a_k and b_k. Matching $\frac{1}{2}\operatorname{tr} M_m(x)$ to (6) (with known coefficients from Section 2), gives m equations. Solutions lie on m-manifolds, but we found that there was no harm in imposing the symmetry [5,7]

$$a_k = b_{m+1-k}, \quad k = 1, \ldots, m, \tag{10}$$

which leads to isolated real solutions. The above symmetry leads to $K(-x) = K(x)^{\mathrm{T}}$.

(One can also consider another symmetry, often known as time-symmetry, namely $K(x) = K(-x)^{-1}$ [3,12]. This can be imposed instead of (10). It turns out that $(m+1)$-stage time-symmetric methods can have an $a_k = 0$ or $b_k = 0$ so that the $(m+1)$-stage method is comparable in effort to an m-stage method of the form (10). It seems that the best m-stage solutions with (10) are comparable to the best $(m+1)$-stage time-symmetric solutions, although we have not explored this point in detail.)

As with the determination of the stability polynomial coefficients, some care must be exercised in solving the polynomial equations, particularly for $m \geqslant 6$. For a given m, n combination, there can be several isolated solutions. We choose those with smallest $\max |b_k|$, i.e., solutions with relatively small time steps between stages, for further study. Table 3 lists several interesting kernels we have found in this manner.

We would like to draw the reader's particular attention to the method with $m = 3$ stages and order $n = 4$. This is a very stable and accurate method with very simple coefficients ($\pm 1/\sqrt{24}$ and 1) which we have found very useful in applications (see Section 5).

4. Determination of correctors

Now that we have determined several kernels K suitable for use in a symplectic corrector algorithm, we move on to illustrate one approach to finding correctors C. For simplicity, we choose C to also be of the form of an m_c-stage composition method: for $k = 1, \ldots, m_c$,

Table 3
Selected m-stage, nth order kernels. Only the m b_k values are listed since $a_k = b_{m+1-k}$. The stability x^* and relative minimum effort m/x^* are also listed for reference

n	m	b_k	x^*	m/x^*
4	3	$1/\sqrt{24}$, $-1/\sqrt{24}$, 1	5.695	0.527
6	3	−0.1591975399846911	2.752	1.090
		0.5098105229662746		
		0.6493870170184167		
6	5	0.308242839898840	6.762	0.739
		0.193663909022838		
		−0.112814793505401		
		0.317306586332861		
		0.293601458250863		
8	4	−0.1105491302584067	4.635	0.863
		0.927118860716406		
		−0.1860052895683256		
		0.3694355591103265		
8	5	−0.0569276507744276	6.636	0.753
		−0.3591147800584599		
		0.1709468364771231		
		1.070147265470851		
		0.1749483288849129		

$$\begin{aligned} p_k &= p_{k-1} - \tau B_k \nabla V(q_{k-1}), \\ q_k &= q_{k-1} + \tau A_k \nabla T(p_k). \end{aligned} \tag{11}$$

This has two advantages. First, C^{-1} is then also an explicit method of the same form as above. (It is merely the reverse.) The second advantage is that then the full symplectic corrector algorithm $C^{-1}KC$ may also be re-expressed as an $(m+2m_c)$-stage splitting method [2], so that relatively minor modifications of our codes for finding K may be employed. Note, of course, that the kernel coefficients will be fixed at their (optimal) values, and only the $2m_c$ corrector coefficients are unknown. (We do not force any symmetries on the coefficients of the correctors.)

In related work, Gray and Manolopoulos [2] carried out some very similar manipulations, and fit the corresponding M_{m+2m_c} to the exact evolution operator M to various orders n. Another approach,

Table 4
Correctors for two of our n, m kernels. In both cases we chose $m_c = 2$ so that only four parameters define each corrector

n	m	A_k	B_k
4	3	0.4597304329110349	0.1701630351972254
		0.1513593913470135	0.532678498596960
6	5	$0.965578471522036 \cdot 10^{-2}$	0.382873441436599
		0.214834477794471	−0.309293457591468

which is simpler in several ways, is to recall property (M4) above and fit $R = M^{\mathrm{T}}M \approx I$. Essentially, this uses the fact that any elliptic area-preserving map of the plane can be conjugated by a change of variables to a rotation. The change of variables we seek is the corrector, C. The elements of R are polynomials in x. Since the overall method is symplectic ($\det M_{m+2m_c} = 1$), and has the correct trace to order n, one need only fit R_{11} to 1 and R_{12} to 0 to some order n_R in x, with the other two components then automatically satisfying the order conditions. The details of this procedure are straightforward and similar in spirit to fitting to the components of M [2].

There is considerable flexibility in defining correctors C. One must decide on a suitable number of corrector stages, m_c, and the order to which the method will be approximately orthogonal, n_R. For simplicity, we choose m_c and n_R to give isolated solutions. If we find several solutions for a given m_c and n_R, we take the one which minimizes $|R(x) - I|$ over the whole interval $0 < x < x^*$.

Table 4 lists correctors found for our $m = 3$, $n = 4$ and $m = 5$, $n = 6$ kernels. In both cases we chose $m_c = 2$, which then turns out to imply that $n_R = 5$ for the corrector solutions to be isolated. Considerably more work could be devoted to choosing correctors for these kernels. Even without this, we obtained excellent accuracies (and of course stabilities) in our test application in Section 5.

5. Application: discretized time-dependent Schrödinger equation

For development of the stability polynomials of interest, it suffices to consider the linear Hamiltonian system of $2N$ equations,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}p &= -Hq, \\ \frac{\mathrm{d}}{\mathrm{d}t}q &= Hp, \end{aligned} \tag{12}$$

which is generated from the (conserved) Hamiltonian

$$h(q,p) = \tfrac{1}{2}\left(p^{\mathrm{T}}Hp + q^{\mathrm{T}}Hq\right). \tag{13}$$

The column vectors q and p each have N components, and H is assumed to be an $N \times N$ real symmetric matrix. (The discretized time-dependent Schrödinger equation can be written in the above form [2].)

A splitting method solution, (2), in this case is of the form, for $k = 1, \ldots, m$,

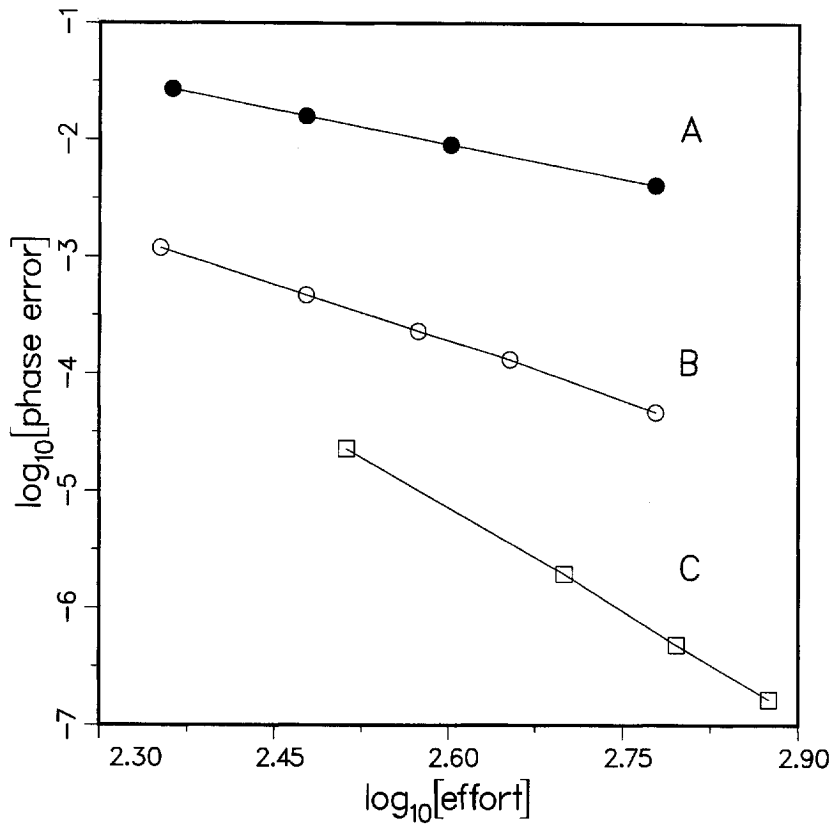

Fig. 2. Accuracy/effort plots associated with the test problem of [2]. Effort is taken to be the number of stages, m, multiplied by the number of time steps to propagate over a time period τ_e. The accuracy measure is the root mean square error associated with a time correlation function, and is a measure of the phase error. Curve A corresponds to the usual leapfrog method, curve B is our $m = 3$, $n = 4$ symplectic corrector and curve C is our $m = 5$, $n = 6$ symplectic corrector. (The leapfrog results, of course, are calculated by assuming an effective stage number of one.)

$$\begin{aligned} p_k &= p_{k-1} - \tau b_k H q_{k-1}, \\ q_k &= q_{k-1} + \tau a_k H p_k. \end{aligned} \tag{14}$$

The stability limit will be $\tau < x^*/|H|_{\max}$, where $|H|_{\max}$ is the maximum absolute magnitude eigenvalue of H.

As discussed in detail in [2], the discretized time-dependent Schrödinger equation,

$$\mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t}\psi(t) = H\psi(t), \tag{15}$$

is equivalent to the Hamiltonian system $h = \frac{1}{2}(p^{\mathrm{T}}Hp + q^{\mathrm{T}}Hq)$ when one identifies $\mathrm{Re}\,\psi$ with q and $\mathrm{Re}\,\psi$ with p. H is an $N \times N$ symmetric matrix that results from the discretization of the underlying partial differential equation. We implemented the test problem given in [2], which is a relatively small problem with $N = 128$ that results when a particle in a certain one-dimensional potential problem is discretized.

We considered the time evolution of an initial condition corresponding to a Gaussian function displaced from the equilibrium position of the potential [2]. The resulting wave packet $\psi(t)$ then exhibits beat frequencies consistent with the twenty-four bound states in the problem. A measure of the phase error is obtained by considering the correlation function, $S(t) = \langle \psi(0) \mid \psi(t) \rangle$, and calculating the corresponding root mean square error relative to a completely converged calculation over a time span from 0 to $50\tau_e$, where τ_e is the period of oscillation associated with the harmonic approximation to the potential [2]. Fig. 2 displays the corresponding phase error calculated when the above equation is solved with the leapfrog method, the $m = 3,\ n = 4$ and the $m = 5,\ n = 6$ symplectic correctors defined in Tables 3 and 4. The $m = 3,\ n = 4$ symplectic corrector has a minimum effort on the scale shown that is almost indistinguishable from that of leapfrog, yet provides more than an order of improvement in accuracy. Actually, the $n = 4$ method of López-Marcos et al. [3] gives almost identical performance in this example, and the $m = 3,\ n = 4$ symplectic corrector of Gray and Manolopoulos [2], while not optimal, is also similar. Our work shows, however, that higher orders and significantly greater accuracy can be achieved with surprisingly modest increases in effort. For example, Fig. 2 also shows our $m = 5,\ n = 6$ symplectic corrector results, which show a further order of magnitude improvement.

Acknowledgements

We are both indebted to Drs. J.G. Verwer and J.M. Sanz-Serna for organizing the Workshop on Innovative Time Integrators, held at the CWI in Amsterdam (October 30–November 1, 1996), that led to this contribution. S.K.G. was supported by the Office of Basic Energy Sciences, Division of Chemical Sciences, U.S. Department of Energy under Contract No. W-31-109-ENG-38. We would also like to thank two referees for their careful work.

References

[1] J.C. Butcher, The effective order of Runge–Kutta methods, Lecture Notes in Mathematics 109 (Springer, New York, 1969) 133–139.

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

[3] M.A. López-Marcos, J.M. Sanz-Serna and R.D. Skeel, An explicit symplectic integrator with maximal stability interval, in: D.F. Griffiths and G.A. Watson, eds., *Numerical Analysis, A.R. Mitchell 75th Birthday Volume* (World Scientific, Singapore, 1996) 163–176.

[4] M.A. López-Marcos, J.M. Sanz-Serna and R.D. Skeel, Cheap enhancement of symplectic correctors, in: D.F. Griffiths and G.A. Watson, eds., *Numerical Analysis 1995* (Longman, 1996) 107–122.

[5] R.I. McLachlan, On the numerical integration of ordinary differential equations by symmetric composition methods, *SIAM J. Sci. Comput.* 16 (1995) 151–168.

[6] R.I. McLachlan, More on symplectic correctors, in: J. Marsden, G. Patrick and W.F. Shadwick, eds., *Integration Algorithms for Classical Mechanics* (The Fields Institute, AMS, 1996).

[7] R.I. McLachlan and P. Atela, The accuracy of symplectic integrators, *Nonlinearity* 5 (1992) 541–562.

[8] R.I. McLachlan and G.R.W. Quispel, Generating functions for dynamical systems with symmetries, integrals, and differential invariants, *Physica D*, submitted.

[9] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 2nd ed., 1992).
[10] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, London, 1994).
[11] J. Wisdom, M. Holman and J. Touma, Symplectic correctors, in: J. Marsden, G. Patrick and W.F. Shadwick, eds., *Integration Algorithms for Classical Mechanics* (The Fields Institute, AMS, 1996).
[12] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett. A* 150 (1990) 262–268.