

Forward and non-forward symplectic integrators in solving classical dynamics problems

SIU A. CHIN*

Department of Physics, Texas A&M University, College Station, TX 77843, USA

(Received 19 March 2007; revised version received 18 April 2007; accepted 27 April 2007)

Forward time step integrators are splitting algorithms with only positive splitting coefficients. When used in solving physical evolution equations, these positive coefficients correspond to positive time steps. Forward algorithms are essential for solving time-irreversible equations that cannot be evolved using backward time steps. However, forward integrators are also better in solving time-reversible equations of classical dynamics by tracking as closely as possible the physical trajectory. This work compares in detail various forward and non-forward fourth-order integrators using three, four, five and six force evaluations. In the case of solving the 2D Kepler orbit, all non-forward integrators are optimized by simply minimizing the size of their backward time steps

Keywords: Symplectic integrators; Splitting methods; Forward factorization algorithms; Keplerian orbits

AMS Subject Classifications: 65P10; 70H15; 70F15; 65Z05

1. Introduction

Many physical evolution equations are of the form

$$\frac{\partial w}{\partial t} = (T + V)w, \quad (1)$$

where T and V are noncommuting operators. Important examples include the imaginary time Schrödinger equation

$$\frac{\partial \psi}{\partial \tau} = \left(\frac{1}{2} \nabla^2 - V \right) \psi \quad (2)$$

and the Fokker-Planck equation

$$\frac{\partial}{\partial t} \rho(\mathbf{x}, t) = \frac{1}{2} \nabla^2 \rho(\mathbf{x}, t) - \nabla \cdot [\mathbf{v}(\mathbf{x}) \rho(\mathbf{x}, t)]. \quad (3)$$

Because the diffusion kernel ∇^2 cannot be evolved backward in time, both of these are *time-irreversible* evolution equations. Aside from these obvious examples of (1), any pair

*Email: chin@physics.tamu.edu

of equations of the form

$$\frac{d\mathbf{q}}{dt} = \mathbf{v}(\mathbf{p}), \quad \frac{d\mathbf{p}}{dt} = \mathbf{F}(\mathbf{q}), \quad (4)$$

can also be casted into the form (1). This is because the evolution of a general function $W(\mathbf{p}, \mathbf{q})$ (including \mathbf{q} and \mathbf{p} themselves) can be formulated as

$$\begin{aligned} \frac{dW}{dt} &= \frac{\partial W}{\partial \mathbf{q}} \cdot \frac{d\mathbf{q}}{dt} + \frac{\partial W}{\partial \mathbf{p}} \cdot \frac{d\mathbf{p}}{dt} \\ &= \left(\mathbf{v}(\mathbf{p}) \cdot \frac{\partial}{\partial \mathbf{q}} + \mathbf{F}(\mathbf{q}) \cdot \frac{\partial}{\partial \mathbf{p}} \right) W, \end{aligned} \quad (5)$$

from which one can identify

$$T = \mathbf{v}(\mathbf{p}) \cdot \frac{\partial}{\partial \mathbf{q}} \quad \text{and} \quad V = \mathbf{F}(\mathbf{q}) \cdot \frac{\partial}{\partial \mathbf{p}}. \quad (6)$$

Classical Hamiltonian dynamics corresponds to

$$\mathbf{v}(\mathbf{p}) = \frac{\mathbf{p}}{m} \quad (7)$$

and the resulting evolution (4) is *time-reversible*.

The generic evolution equation (1) can be solved iteratively

$$w(t + \varepsilon) = e^{\varepsilon(T+V)} w(t) \quad (8)$$

by approximating the short time evolution operator $e^{\varepsilon(T+V)}$ to any order in ε via

$$e^{\varepsilon(T+V)} = \prod_{i=1}^N e^{t_i \varepsilon T} e^{v_i \varepsilon V}, \quad (9)$$

assuming that the effect of $e^{\varepsilon T}$ and $e^{\varepsilon V}$ can be computed exactly. The set of coefficients $\{t_i, v_i\}$ are determined by the order condition. For classical dynamics (6), every factorizations of the form (9) produces a symplectic integrator [1–10] which is an ordered sequence of alternating displacements of \mathbf{p} and \mathbf{q} preserving Poincaré invariants. For periodic motion, their energy errors are bounded and periodic, in contrast to explicit Runge-Kutta type algorithms whose energy error grows linearly with the number of periods [11, 12]. However, for solving time-irreversible evolution equations such as (2) or (3) with $T = \nabla^2/2$, the Green's function

$$G(\mathbf{r}', \mathbf{r}; t_i \varepsilon) \propto e^{-(\mathbf{r}' - \mathbf{r})^2 / (2t_i \varepsilon)} \quad (10)$$

is the diffusion kernel only if t_i is positive. If t_i were negative, then the kernel is unbound and there is no way of simulating the diffusion process backward in time.

Historically, symplectic integrators were developed extensively for use in classical dynamics [1–10]. Since classical dynamics is time-reversible, there was no impetus for demanding that all t_i be positive. Moreover, Sheng [13] and Suzuki [14] have proved that all factorizations of the form (9) beyond second order must contain some negative coefficients in the set $\{t_i, v_i\}$. Goldman and Kaper [15] later proved that beyond second order, *both* operators must have at least one negative coefficient. Thus all conventional splitting schemes beyond second order must contain some negative coefficients and none can be used to solve time-irreversible problems. Because of the Sheng-Suzuki theorem, it is also difficult to see how one can devise all positive coefficients, *forward* time-step algorithms.

The operator product (9) has the general Campbell-Baker-Hausdorff expansion,

$$\prod_{i=1}^N e^{t_i \varepsilon T} e^{v_i \varepsilon V} = \exp \varepsilon (e_T T + e_V V + \varepsilon e_{TV} [T, V] + \varepsilon^2 e_{TTV} [T, [T, V]] + \varepsilon^2 e_{VTV} [V, [T, V]] + \dots) \tag{11}$$

where all the error coefficients e_T , e_{TV} , e_{VTV} , etc., are calculable functions of $\{t_i, v_i\}$, in particular,

$$e_T = \sum_{i=1}^N t_i \quad \text{and} \quad e_V = \sum_{i=1}^N v_i. \tag{12}$$

In order for the product to be consistent with the original evolution operator, the coefficients $\{t_i, v_i\}$ must satisfy the above constraints with $e_T = 1$ and $e_V = 1$. Forcing the remaining error coefficients to vanish results in order conditions that $\{t_i, v_i\}$ must satisfy. It is easy to force $e_{TV} = 0$. Any left-right symmetric product will do. For example,

$$\mathcal{T}_2(\varepsilon) = e^{(1/2)\varepsilon T} e^{\varepsilon V} e^{(1/2)\varepsilon T} = e^{\varepsilon(T+V - (1/24)\varepsilon^2[T, [T, V]] + (1/12)\varepsilon^2[V, [V, T]] + \dots)} \tag{13}$$

produces the following second order symplectic algorithm according to (6):

$$\begin{aligned} \mathbf{q}_1 &= \mathbf{q}_0 + \frac{1}{2} \varepsilon \frac{\mathbf{p}_0}{m} \\ \mathbf{p}_1 &= \mathbf{p}_0 + \varepsilon \mathbf{F}(\mathbf{q}_1) \\ \mathbf{q}_2 &= \mathbf{q}_1 + \frac{1}{2} \varepsilon \frac{\mathbf{p}_1}{m} \end{aligned} \tag{14}$$

where the last numbered variables are the updated variables. Thus any symmetric splitting with $e_T = e_V = 1$ will result in at least a second order algorithm. The surprise is that, as first shown by Sheng [13], beyond second order a general sum of products of the form (11) is incompatible with having positive coefficients. More specifically, Suzuki [14] shown that the two error coefficients e_{TTV} and e_{VTV} cannot both be forced to zero for positive coefficients $\{t_i, v_i\}$. Since Takahashi and Imada [16] have shown that $[V, [T, V]] = |\nabla V(r)|^2$ is a local potential function when solving the imaginary time Schrödinger equation, Suzuki suggested [17] that this error commutator be kept and ways be found to eliminate $[T, [T, V]]$.

Following up on Suzuki’s suggestion, this author derived three simple fourth-order forward algorithms [18] in 1997 and demonstrated their efficiency in solving Kepler’s orbit. Interestingly, it was found that classically $[V, [T, V]]$ produces a force (also with potential $|\nabla V(r)|^2$) first derived by Ruth [1] via canonical transformations. The two forward schemes *A* and *B* derived in [18] were also known to Suzuki [19] based on McLachan’s result [20] on slightly perturbed Hamiltonians. However, Suzuki did not implement them to do any calculation.

Since 1997, fourth-order forward algorithms have been widely applied to time-irreversible systems such as the Fokker-Planck equation in deriving the first fourth-order Langevin algorithm [21], the Kramers equation [22] for describing stochastic dynamics, the Diffusion Monte Carlo algorithm [22, 23] for solving quantum many-body ground states, the grid based imaginary time Schrödinger [24] equation in doing density functional calculations, Path-Integral-Monte Carlo methods [25–30] for computing the quantum trace at finite temperature, short time evolved wave functions [31] for doing variational quantum many-body calculations, the Gross-Pitaevskii equation for describing a Bose-Einstein condensate in a

rotating anisotropic trap [32] and electrons in a magnetic field confined by quantum dots [33] and rings [34]. These forward fourth-order algorithms are far more accurate than second order algorithms and allow very large step sizes to be used.

While forward algorithms are indispensable for solving time-irreversible equations, from their inception [18] they have also been shown to be efficient in solving time-reversible equations. In comparison with explicit Runge-Kutta algorithms and conventional non-forward symplectic integrators, forward algorithms have been shown to be superior in solving the Kepler problem [18, 35, 36], gravitational few-body problems [37], the real time Schrödinger equation [38], the real time Schrödinger equation in a laser field [39–41] and specially the radial Schrödinger equation [42]. The incorporation of the commutator $[V, [T, V]]$ in forward algorithms has also inspired a new class of higher order *gradient* symplectic integrators [39, 43–45]. These new algorithms, through not forward beyond fourth-order, have less backward steps at higher orders.

The reason why forward integrators are also better in solving time-reversible, classical dynamics problems is not well understood. From the perspective of the operator product approximation (9), as long as the second-order error terms are zero, any symmetric factorization scheme will be fourth-order; it should not matter whether these error terms are forced to zero with positive or negative coefficients. However, just as in the discussion of time-reversible and time-irreversible algorithms, one must move beyond the purely algebraic discussion of factorization schemes to examine how the resulting algorithms produce the solution of any particular equation. This work uncovers a crucial difference between forward and non-forward schemes when they are implemented as integrators of classical dynamics problems. For classical dynamics, if the trajectory is the exact solution to (4), then the force is evaluated *only* along the trajectory. As will be shown below, non-forward algorithms, when compared to forward algorithms at the same time step size ε , evaluate the force at intermediate positions far from the trajectory. As one reduces the size of the negative time steps, one also reduces the distance of these force evaluation points from the trajectory. In all cases examined, non-forward algorithms are improved by simply reducing the size of their backward time steps, allowing the force to be evaluated more closely along the trajectory. One can argue that these intermediate force evaluation points are not the trajectory outputs of the integrator and their placements are not required to be on the trajectory. That is correct. However, since the exact trajectory is determined *only* by forces evaluated on the trajectory, any unnecessary force evaluation off the trajectory is just ‘wasteful’. One then must reduce the time step to bring the force evaluation points closer to the trajectory. This is exactly what is observed in the following comparisons. To achieve the same accuracy, non-forward integrators must use smaller time steps. The question here is not about correctness; it is about efficiency. In the following sections we will compare in detail both types of integrators with three, four, five and six force evaluations.

2. Integrators with three force evaluations

We will begin by comparing the efficiency of integrators in solving the 2D Kepler problem defined by the Hamiltonian

$$H = \frac{1}{2}\mathbf{p}^2 - \frac{1}{q}. \quad (15)$$

(Henceforth, we will always normalize the Hamiltonian with kinetic energy $\mathbf{p}^2/2$ and $m = 1$.) This problem is an excellent benchmark because one can gauge an integrator’s performance

not just by its energy error but also by its orbital precession error [12, 35, 36]. The latter is a more direct measure of the accuracy of the computed orbit.

There are basically three fourth-order integrators that required only three force evaluations:

1) The non-forward Forest-Ruth (FR) integrator [3–5],

$$\mathcal{T}_{FR}(\varepsilon) = \mathcal{T}_2(a_1\varepsilon)\mathcal{T}_2(a_0\varepsilon)\mathcal{T}_2(a_1\varepsilon) \tag{16}$$

where

$$a_1 = \frac{1}{2 - 2^{1/3}} \approx 1.35, \quad a_0 = -\frac{2^{1/3}}{2 - 2^{1/3}} \approx -1.70. \tag{17}$$

2) The forward integrator A [18, 19]:

$$\mathcal{T}_A(\varepsilon) \equiv e^{(1/6)\varepsilon V} e^{(1/2)\varepsilon T} e^{(2/3)\varepsilon \tilde{V}} e^{(1/2)\varepsilon T} e^{(1/6)\varepsilon V}, \tag{18}$$

with \tilde{V} defined by

$$\tilde{V} = V + \frac{1}{48}\varepsilon^2[V, [T, V]] \tag{19}$$

corresponding to an effective force

$$\tilde{\mathbf{F}}(\mathbf{q}) = \mathbf{F}(\mathbf{q}) + \frac{1}{48}\varepsilon^2\nabla(|\mathbf{F}(\mathbf{q})|^2). \tag{20}$$

Transcribing each operator in (18) yields the integrator

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{p}_0 + \frac{1}{6}\varepsilon \mathbf{F}(\mathbf{q}_0) \\ \mathbf{q}_1 &= \mathbf{q}_0 + \frac{1}{2}\varepsilon \mathbf{p}_1 \\ \mathbf{p}_2 &= \mathbf{p}_1 + \frac{2}{3}\varepsilon \tilde{\mathbf{F}}(\mathbf{q}_1) \\ \mathbf{q}_2 &= \mathbf{q}_1 + \frac{1}{2}\varepsilon \mathbf{p}_2 \\ \mathbf{p}_3 &= \mathbf{p}_2 + \frac{1}{6}\varepsilon \mathbf{F}(\mathbf{q}_2). \end{aligned} \tag{21}$$

Starting with initial values \mathbf{p}_0 and \mathbf{q}_0 , the updated variables are $\mathbf{p} = \mathbf{p}_3$ and $\mathbf{q} = \mathbf{q}_2$. Algorithm A only requires two evaluations of the force and one evaluation of the force gradient. Recently, Omelyan [45] has suggested that the effective force (20) can be evaluated by extrapolation

$$\tilde{\mathbf{F}}(\mathbf{q}) = \mathbf{F}\left(\mathbf{q} + \frac{1}{24}\varepsilon^2\mathbf{F}(\mathbf{q})\right) + O(\varepsilon^4). \tag{22}$$

The resulting integrator remains angular momentum and phase-volume conserving and is nearly indistinguishable from a fully symplectic integrator when solving the Kepler problem. We will denote this use of an extrapolated effective force (22) as algorithm A'. Algorithm A' only requires three force evaluations.

3) The Runge-Kutta-Nystrom (RKN) integrator [46]. If one expresses $\mathbf{p} = \mathbf{p}_3$ and $\mathbf{q} = \mathbf{q}_2$ directly in terms of \mathbf{p}_0 and \mathbf{q}_0 , then (21) reduces to

$$\begin{aligned} \mathbf{q} &= \mathbf{q}_0 + \varepsilon \mathbf{p}_0 + \frac{1}{6}\varepsilon^2(\mathbf{F}_0 + 2\mathbf{F}_1) \\ \mathbf{p} &= \mathbf{p}_0 + \frac{1}{6}\varepsilon(\mathbf{F}_0 + 4\mathbf{F}_1 + \mathbf{F}_2). \end{aligned} \tag{23}$$

This is the form of the RKN algorithm, with $\mathbf{F}_0 = \mathbf{F}(\mathbf{q}_0)$, $\mathbf{F}_2 = \mathbf{F}(\mathbf{q}_2)$, but with $\mathbf{F}_1 = \tilde{\mathbf{F}}(\mathbf{q}_1)$. The conventional RKN algorithm is defined by $\mathbf{F}_0 = \mathbf{F}(\mathbf{q}_0)$, $\mathbf{F}_1 = \mathbf{F}(\mathbf{q}'_1)$ and $\mathbf{F}_2 = \mathbf{F}(\mathbf{q}'_2)$, where

$$\begin{aligned}\mathbf{q}'_1 &= \mathbf{q}_0 + \frac{\varepsilon}{2}\mathbf{p}_0 + \frac{1}{2}\left(\frac{\varepsilon}{2}\right)^2\mathbf{F}_0 \\ \mathbf{q}'_2 &= \mathbf{q}_0 + \varepsilon\mathbf{p}_0 + \frac{1}{2}\varepsilon^2\mathbf{F}(\mathbf{q}'_1)\end{aligned}\quad (24)$$

are the estimated midpoint and final position respectively.

Figure A1 (see Appendix A) shows the fourth-order energy error coefficients of these four algorithms at step size $\varepsilon = P/5000$, where P is the period of an highly eccentric orbit with initial values $\mathbf{q} = (10, 0)$, $\mathbf{p} = (0, 1/10)$ and eccentricity $e = 0.9$. The error coefficient is obtained by dividing the energy error by ε^4 at smaller and smaller ε until a convergent curve emerges independent of ε . The curve is further normalized by the initial energy. Thus each algorithm has a characteristic error coefficient, its error ‘fingerprint’, in solving the Kepler orbit. For symplectic algorithms, this convergence is already set in when $\varepsilon \approx P/1000$. For the RKN algorithm, the energy error curve after the mid period keep on lowering with decreasing step size, showing no sign of convergence at finite ε . The error only spikes at mid period near the pericenter point. Non-symplectic integrator such RKN are characterized by an irreversible increase in the energy error after each period. The error spike of the FR algorithm is nearly ten times as large as that of algorithm A. The extrapolated gradient algorithm A' closely matches that of A.

Since all symplectic integrators have periodic energy errors, the energy error is not the most critical benchmark. The orbital precession error, as measured by the rotation of Laplace–Runge–Lenz (LRL) vector [35, 36, 47], is more discriminating. Figure A2 shows the rotation angle of the LRL vector along the trajectory of the particle. The corresponding error coefficient is again extracted by dividing by ε^4 . In this case, the RKN integrator shows the same convergence as the symplectic integrators. Thus the precession error coefficient is well-defined and irreversible for all algorithms. The FR integrator’s error is three times as large as that of RKN and 10 times as large as integrator A and A' . When the force gradient is extrapolated, the energy error remains periodic, but the precession error can differs substantially. Here A' ’s error is smaller, but in other cases it may not be. We will revisit this point later.

In order to understand the poor performance of the FR integrator, we track its approach toward the pericenter point 3 at two time steps earlier at point 1, as shown in figure A3. The time step used here is $\varepsilon = P/400$. The FR integrator consists of three applications of $\mathcal{T}_2(\varepsilon)$, resulting in three overlapping triangles. The first application of $\mathcal{T}_2(a_1\varepsilon)$ begins at position 1, evaluates the force at F1, and lands at 1a. This is the triangle 1-F1-1a. The application of the backward substep $\mathcal{T}_2(a_0\varepsilon)$ begins at 1a, evaluates the force at F2 and brings trajectory back past the starting point to 1b. This is the backward triangle 1a-F2-1b. The final application of $\mathcal{T}_2(a_1\varepsilon)$ begins at 1b, evaluates the force at F3 and lands the trajectory at position 2. This is the final triangle 1b-F3-2. Starting at position 2, the algorithm repeats its three overlapping triangles and zigzags its way to point 3. It is remarkable that FR can achieve fourth-order accuracy by such a tremendous zigzagging. Notice that as the FR algorithm tries to turn the ‘corner’ near the pericenter 3, all of its force evaluation points are *far off* the trajectory. In figure A4, the positions where each algorithm calculates the force are plotted. The backward loops executed by FR far off the trajectory is conspicuous. This is the fundamental reason by all non-forward algorithms perform poorly. By comparison, forward algorithm A always evaluate the force and the force-gradient close to the exact trajectory. Since RKN is similar in form to A, it strays from the exact trajectory only near the pericenter point 3.

3. Integrators with four force evaluations

Because the error of the FR integrator is uncomfortably large, there is an ongoing effort to construct better non-forward algorithms by use of more force evaluations. A non-forward fourth-order algorithm can be obtained by generalizing (16) to

$$\mathcal{T}_4(\varepsilon) = \prod_{i=1}^N \mathcal{T}_2(a_i \varepsilon), \tag{25}$$

provided that the coefficients a_i are left-right symmetric satisfying [3, 5, 48]

$$\sum_{i=1}^N a_i = 1 \quad \text{and} \quad \sum_{i=1}^N a_i^3 = 0. \tag{26}$$

Unfortunately, for $N = 4$, there are no real solutions to the above equations. We will examine algorithms of the general form

$$\mathcal{T}_{M1} = \cdots \exp(\varepsilon t_0 T) \exp(\varepsilon v_1 V) \exp(\varepsilon t_1 T) \exp(\varepsilon v_2 V) \exp(\varepsilon t_2 T), \tag{27}$$

previously studied by McLachlan’s [49]. Since the algorithm is left-right symmetric, only operators from the center to the right are indicated. For a fourth-order integrator, the order condition requires that

$$v_1 = \frac{1}{2} - v_2, \quad t_2 = \frac{1}{6} - 4t_1 v_1^2, \quad t_0 = 1 - 2(t_1 + t_2), \tag{28}$$

$$w = \sqrt{3 - 12t_1 + 9t_1^2}, \quad v_2 = \frac{1}{4} \left(1 \mp \sqrt{\frac{9t_1 - 4 \pm 2w}{3t_1}} \right) \tag{29}$$

and that the free parameter $t_1 < 0$. There are four solution branches for v_2 . The choice of

$$t_1 = \frac{121}{3924} (12 - \sqrt{471}) \approx -0.299$$

with

$$v_2 = \frac{1}{4} \left(1 + \sqrt{\frac{9t_1 - 4 + 2w}{3t_1}} \right) \tag{30}$$

reproduces McLachlan’s [49] recommended algorithm (denote here as algorithm M). By simply reducing the size of the negative time steps t_1 , we obtain results as shown in figure A5 and figure A6. The choice of $t_1 = -1/24$ yielded simple analytical coefficients

$$v_2 = \frac{1 + \sqrt{5}}{4} \quad \text{and} \quad t_2 = \frac{11 - \sqrt{5}}{48}. \tag{31}$$

Also shown are results of forward integrator C [18]:

$$\mathcal{T}_C = \cdots \exp\left(\frac{1}{4}\varepsilon \tilde{V}\right) \exp\left(\frac{1}{3}\varepsilon T\right) \exp\left(\frac{3}{8}\varepsilon V\right) \exp\left(\frac{1}{6}\varepsilon T\right), \tag{32}$$

where \tilde{V} is as defined in (19) with the same interpretation (20). Algorithm C uses three force and one force gradient evaluations. The force gradient can again be extrapolated by another force evaluation. The results are similar and will be omitted in this comparison. To understand

the poor performance of non-forward algorithms, we again plot their force evaluation points in figure A7. Starting at position 2, McLachlan’s algorithm evaluates the force at F1, back tracks and evaluates the force the second time at F2, takes a giant leap to F3, then back track again to F4. By reducing t_1 to $-1/24$, the back tracking steps F2 and F3 are reduced to f2 and f3. However, it is not possible to move any force evaluation points closer to the midpoint of the trajectory. The force evaluation points of algorithm C are indicated by circles. Its first and last force evaluation points nearly coincide with F1 and F4, however, it evaluates the force and the force gradient right at the midpoint of the trajectory as shown by the unobstructed circle.

Better algorithms are obtained by interchanging $T \leftrightarrow V$ in (27)

$$\mathcal{T}_{M2} = \dots \exp(\varepsilon t_0 V) \exp(\varepsilon v_1 T) \exp(\varepsilon t_1 V) \exp(\varepsilon v_2 T) \exp(\varepsilon t_2 V) \tag{33}$$

so that the momentum is updated first with the choice

$$v_2 = \frac{1}{4} \left(1 - \sqrt{\frac{9t_1 - 4 + 2w}{3t_1}} \right). \tag{34}$$

Now the force is evaluated initially, at two intermediate points, and at the midpoint. The results, as shown in figures A8 and A9, are much improved over the previous case. However, the locations of the forces evaluation points remain unusual. As shown in figure A10, for $t_1 = -0.1$, the algorithm first evaluates the force at the starting point 2, backtracks past 2 to evaluate the force at F2, leaps forward to evaluate the force near the midpoint at F3, and shoots past the final point 3 to evaluate the force at F4. Tuning the parameter t_1 more negative to -0.5 reduces the back tracking points from F2 to f2 and F4 to f4 and improves the algorithm. However, as t_1 becomes even more negative, such as $t_1 = -1$ or -2 , those back tracking points bunch up very close to the initial and final points and do not sample the trajectory evenly as algorithm C.

4. Integrators with five and six force evaluations

For $N = 5$, (25) can be solved to give

$$\mathcal{T}_4(\varepsilon) = \dots \mathcal{T}_2(a_0\varepsilon)\mathcal{T}_2(a_1\varepsilon)\mathcal{T}_2(a_2\varepsilon), \tag{35}$$

with free parameter α and coefficients

$$a_2 = \alpha a_1, \quad a_0 = -2^{1/3}(1 + \alpha^3)^{1/3} a_1, \tag{36}$$

$$a_1 = \frac{1}{2(1 + \alpha) - 2^{1/3}(1 + \alpha^3)^{1/3}}. \tag{37}$$

The FR integrator is reproduced with $\alpha = 0$. By introducing a non-vanishing a_2 , one is able to reduce the negative step size a_0 . This is shown in figures A11, A12 and A13 show the energy and the precession error as a function of α . While the energy error height is lowest for $\alpha = 0.5$, the precession error is the smallest at $\alpha = 1$. The latter is related to the fact that the backward step size a_0 is minimized at $\alpha = 1$. The resulting algorithm with

$$a_1 = a_2 = \frac{1}{4 - 4^{1/3}}, \quad a_0 = -\frac{4^{1/3}}{4 - 4^{1/3}}, \tag{38}$$

has long been advocated by Creutz and Gocksch [3], Suzuki [48] and McLachlan [50].

Recently, a fundamental theorem [51] has allowed fourth order forward algorithms to be derived for any number of operators [52]. In particular, one can generalize algorithm A to

$N - 1$ force plus one force-gradient evaluations (or N force evaluations using extrapolation). This is the class of algorithm with uniform splitting coefficients

$$t_i = \frac{1}{N - 1}, \quad v_i = \frac{N - 1}{N(N - 2)}, \quad (39)$$

and where the algorithm begins and ends with a momentum updating step:

$$\mathbf{p}' = \mathbf{p} + \frac{1}{2N}\varepsilon \left(\mathbf{F}(\mathbf{q}) + \frac{1}{24(N - 2)}\varepsilon^2 \nabla (|\mathbf{F}(\mathbf{q})|^2) \right). \quad (40)$$

We will denote this class of algorithm as AN. The energy and precession errors for A5 are as indicated in figures A12 and A13. Algorithm A5's precession error is more than 4 times smaller than that of algorithm C and 200 times smaller than non-forward algorithm (35) at $\alpha = 1$. As shown in figure A14, algorithms (35) again characteristically evaluates the force off the trajectory. As the negative time step a_0 is reduced by increasing $\alpha = 0.3$ to $\alpha = 1.0$, the off-trajectory force evaluation triangle is reduced from F2-F3-F4 to f2-f3-f4.

In figures A15 and A16, we compare the energy and precession error of A6 with that of Blanes and Moan [53] (BM), a widely cited fourth-order integrator with six force evaluations. BM's energy error is comparable to that of algorithm C, but its maximum error height is four times that of A6. For the precession error, BM's error is two orders of magnitude larger than A6 and twenty times larger than C. Included in the comparison is algorithm A6', in which the force gradient is computed via extrapolation. Its energy error is nearly indistinguishable from that of A6, however, its precession error is much larger. Blanes and Moan's integrator is superior among non-forward algorithms because it has only two very small backward time steps, as shown in figure A17. (The momentum updating step first version of the BM integrator is not considered because it has much large errors than the position-first version discussed above.)

5. Conclusions

All approximation methods for solving any evolution equation should emulate its exact solution as much as possible. The efficiency of an algorithm cannot be decided on the basis of factorization schemes, in which only the error coefficients of the error commutators are known. In the past, symplectic integrators have been prized for their excellent conservation properties. However, because of the perceived difficulty in circumventing the Shang-Suzuki theorem, forward integrators were not developed until this decade. In this work, we showed that forward integrators are more attuned to the exact solution by evaluating the force closely on the trajectory. By comparison, non-forward integrators, because of their backward time steps, are constrained to evaluate the force off the trajectory, resulting in the loss of efficiency. In all cases studied, non-forward integrators are improved by simply reducing the size of their negative time steps.

Acknowledgements

This work is supported, in part, by a National Science Foundation grant, No. DMS-0310580.

References

- [1] Ruth, R., 1983, A canonical integration technique. *IEEE Transactions on Nuclear Science*, **30**, 2669–2671.
- [2] Neri, F., *Lie Algebra and Canonical Integration*. Department of Physics, University of Maryland, College Park (preprint).

- [3] Creutz, M. and Gocksch, A., 1989, Higher-order hybrid Monte-Carlo algorithms. *Physical Review Letters*, **63**, 9–12.
- [4] Forest, E. and Ruth, R.D., 1990, 4th-order symplectic integration. *Physica D*, **43**, 105–117.
- [5] Yoshida, H., Construction of higher order symplectic integrators, *Physics Letters*, **A150**, 262–268.
- [6] Wisdom, J. and Holman, M., 1991, Symplectic maps for the N-body problem. *Astronomical Journal*, **102**, 1528–1538.
- [7] Yoshida, H., 1993, Recent progress in the theory and application of symplectic integrators. *Celestial Mechanics and Dynamical Astronomy*, **56**, 27–43.
- [8] Hairer, E., Lubich, C. and Wanner, G., 2002, *Geometric Numerical Integration* (Berlin-New York: Springer-Verlag).
- [9] McLachlan, R.I. and Quispel, G.R.W., 2002, Splitting methods. *Acta Numerica*, **11**, 341–434.
- [10] Leimkuhler, B. and Reich, S., 2004, *Simulating Hamiltonian Dynamics* (Cambridge: Cambridge University Press).
- [11] Kinoshita, H., Yoshida, H. and Nakai, H., 1991, Symplectic integrators and their application to dynamical astronomy. *Celestial Mechanics and Dynamical Astronomy*, **50**, 59–71.
- [12] Gladman, B., Duncan, M. and Candy, J., 1991, Symplectic integrators for long-term integration in celestial mechanics. *Celestial Mechanics and Dynamical Astronomy*, **52**, 221–240.
- [13] Sheng, Q., 1989, Solving linear partial differential equations by exponential splitting. *IMA Journal of Numerical Analysis*, **9**, 199–212.
- [14] Suzuki, M., 1991, General theory of fractal path-integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, **32**, 400–407.
- [15] Goldman, D. and Kaper, T.J., 1996, Nth-order operator splitting schemes and nonreversible systems. *SIAM Journal on Numerical Analysis*, **33**, 349–367.
- [16] Takahashi, M. and Imada, M., 1984, Monte Carlo calculation of quantum systems 2 - higher order correction. *Journal of the Physical Society of Japan*, **53**, 3765–3769.
- [17] Suzuki, M., 1995, Hybrid exponential product formulas for unbounded operators with possible applications to Monte Carlo simulations. *Physics Letters*, **A 201**, 425–428.
- [18] Chin, S.A., 1997, Symplectic integrators from composite operator factorizations. *Physics Letters*, **A226**, 344–348.
- [19] Suzuki, M., 1996, New scheme of hybrid Exponential product formulas with applications to quantum Monte Carlo simulations. In: D. Landau, K. Mon and H. Shuttler (Eds), *Computer Simulation Studies in Condensed Matter Physics VIII* (Berlin: Springer), pp. 1–6.
- [20] McLachlan, R., 1995, Composition methods in the presence of a small parameters. *BIT*, **35**, 258–268.
- [21] Forbert, H.A. and Chin, S.A., 2001, Fourth-order algorithms for solving the multivariable Langevin equation and the Kramers equation. *Physical Review*, **E 63**, 016703.
- [22] Forbert, H.A. and Chin, S.A., 2001, Fourth-order diffusion Monte Carlo algorithms for solving quantum many-body problems. *Physical Review*, **B 63**, 144518.
- [23] Aspuru-Guzik, A. and Lester, W.A., 2005, Quantum Monte Carlo: Theory and application to molecular systems. *Advances in Quantum Chemistry*, **49**, 209.
- [24] Auer, J., Krotscheck, E. and Chin, S.A., 2001, A fourth-order real-space algorithm for solving local Schrödinger equations. *Journal of Chemical Physics*, **115**, 6841–6846.
- [25] Jang, S., Jang, S. and Voth, G., 2001, Applications of higher order composite factorization schemes in imaginary time path integral simulations. *Journal of Chemical Physics*, **115**, 7832–7842.
- [26] Chin, S.A., 2004, Quantum statistical calculations and symplectic corrector algorithms. *Physical Review*, **E 69**, 046118.
- [27] Brualla, L., Sakkos, K., Boronat, J. and Casulleras, J., 2004, Higher order and infinite Trotter-number extrapolations in path integral Monte Carlo. *Journal of Chemical Physics*, **121**, 636–643.
- [28] Cuervo, J.E., Roy, P.N. and Boninsegni, M., 2005, Path integral ground state with a fourth-order propagator: Application to condensed helium. *Journal of Chemical Physics*, **122**, 114504.
- [29] Boninsegni, M., Prokof'ev, N. and Svistunov, B., 2006, Worm algorithm for continuous-space path integral Monte Carlo simulations. *Physical Review Letters*, **96**, 070601.
- [30] Mezzacapo, F. and Boninsegni, M., 2006, Superfluidity and quantum melting of p-H-2 clusters. *Physical Review Letters*, **97**, 045301.
- [31] Ciftja, O. and Chin, S.A., 2003, Short-time-evolved wave functions for solving quantum many-body problems. *Physical Review*, **B 68**, 134510.
- [32] Chin, S.A. and Krotscheck, E., 2005, Fourth-order algorithms for solving the imaginary-time Gross-Pitaevskii equation in a rotating anisotropic trap. *Physical Review*, **E 72**, 036705.
- [33] Aichinger, M., Chin, S.A. and Krotscheck, E., 2005, Fourth-order algorithms for solving local Schrödinger equations in a strong magnetic field. *Computer Physics Communications*, **171**, 197–207.
- [34] Aichinger, M., Chin, S.A., Krotscheck, E. and Räsänen, E., 2006, Effects of geometry and impurities on quantum rings in magnetic fields. *Physical Review*, **B 73**, 195310.
- [35] Chin, S.A. and Kidwell, D.W., 2000, Higher-order force gradient symplectic algorithms. *Physical Review*, **E 62**, 8746.
- [36] Scuro, S.R. and Chin, S.A., 2005, Forward symplectic integrators and the long-time phase error in periodic motions. *Physical Review*, **E 71**, 056703.
- [37] Chin, S.A. and Chen, C.R., 2005, Forward symplectic integrators for solving gravitational few-body problems. *Celestial Mechanics and Dynamical Astronomy*, **91**, 301–322.

- [38] Chin, S.A. and Chen, C.R., 2001, Fourth order gradient symplectic integrator methods for solving the time-dependent Schrödinger equation. *Journal of Chemical Physics*, **114**, 7338.
- [39] Chin, S.A. and Chen, C.R., 2002, Gradient symplectic algorithms for solving the Schrödinger equation with time-dependent potentials. *Journal of Chemical Physics*, **117**, 1409.
- [40] Baye, D., Goldstein, G. and Capel, P., 2003, Fourth-order factorization of the evolution operator for time-dependent potentials. *Physics Letters*, **A 317**, 337.
- [41] Goldstein, G. and Baye, D., 2004, Sixth-order factorization of the evolution operator for time-dependent potentials. *Physical Review*, **E 70**, 056703.
- [42] Chin, S.A. and Anisimov, P., 2006, Gradient Symplectic Algorithms for Solving the Radial Schrodinger Equation. *Journal of Chemical Physics*, **124**, 054106.
- [43] Omelyan, I.P., Mryglod, I.M. and Folk, R., 2002, Construction of high-order force-gradient algorithms for integration of motion in classical and quantum systems. *Physical Review*, **E66**, 026701.
- [44] Omelyan, I.P., Mryglod, I.M. and Folk, R., 2003, Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations. *Computer Physics Communications*, **151**, 272.
- [45] Omelyan, I.P., 2006, Extrapolated gradientlike algorithms for molecular dynamics and celestial mechanics simulations. *Physical Review*, **E 74**, 036703.
- [46] Battin, Richard H., 1999, *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised Edition (Reston, VA: AIAA).
- [47] Chin, S.A., 2007, Physics of symplectic integrators: perihelion advances and symplectic corrector algorithms. *Physical Review*, **E 75**, 036701.
- [48] Suzuki, M., 1990, Hybrid exponential product formulas for unbounded operators with possible applications to Monte Carlo simulations. *Physics Letters*, **A 146**, 319–323.
- [49] McLachlan, R.I., 1995, On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM Journal on Scientific Computing*, **16**, 151–168.
- [50] McLachlan, R.I., 2002, Families of high-order composition methods. *Numerical Algorithms*, **31**, 233–246.
- [51] Chin, S.A., 2006, A fundamental theorem on the structure of symplectic integrators. *Physics Letters*, **A 354**, 373–376.
- [52] Chin, S.A., 2006, The complete characterization of fourth-order symplectic integrators with extended-linear coefficients. *Physical Review*, **E 73**, 026705.
- [53] Blanes, S. and Moan, P.C., 2002, Practical symplectic partition Runge-Kutta methods and Runge-Kutta Nyström methods. *Journal of Computational and Applied Mathematics*, **142**, 313–330.

Appendix A: Figures

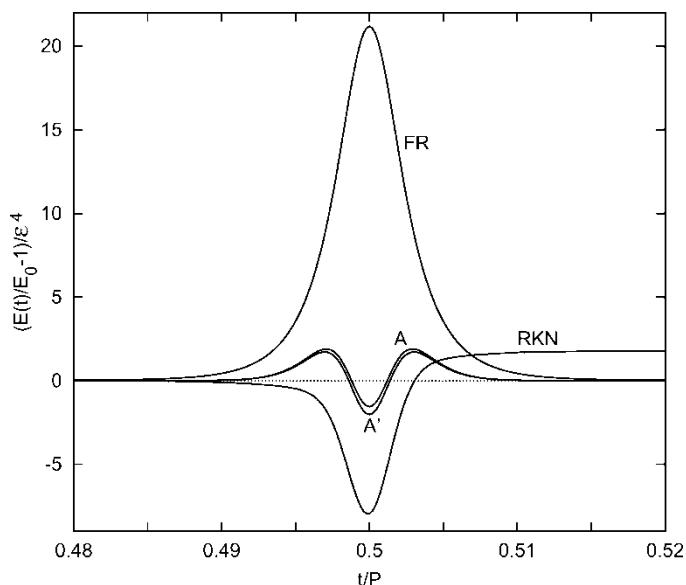


Figure A1. The fourth-order energy error coefficients of forward symplectic integrator A, extrapolated gradient algorithm A', non-forward symplectic integrator FR (Forest-Ruth) and non-symplectic integrator RKN (Runge-Kutta-Nyström), as a function of time in terms of the orbital period P when solving the 2D Kepler orbit. The time step size is denoted by ϵ .

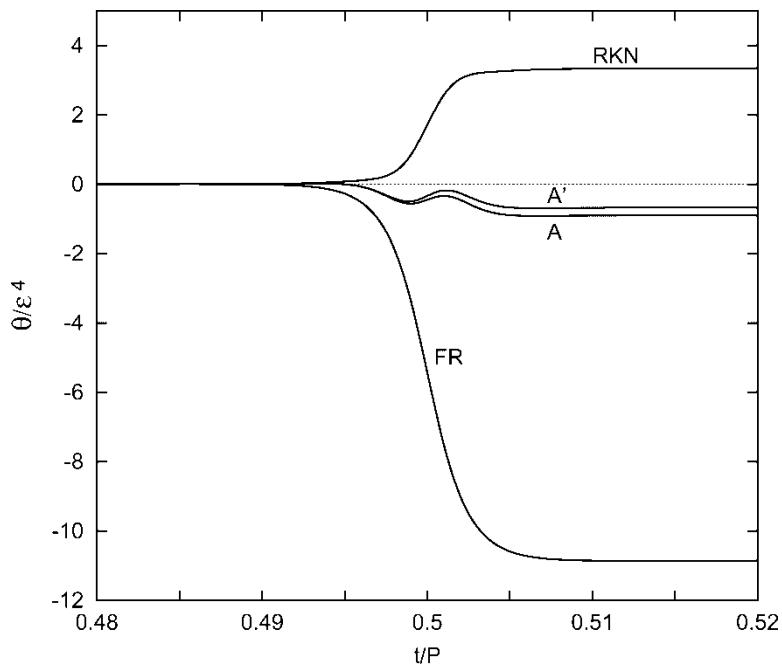


Figure A2. The fourth-order orbital precession error coefficient as measured by the rotation angle of the Laplace-Runge-Lenz (LRL) vector for integrators described in figure A1.

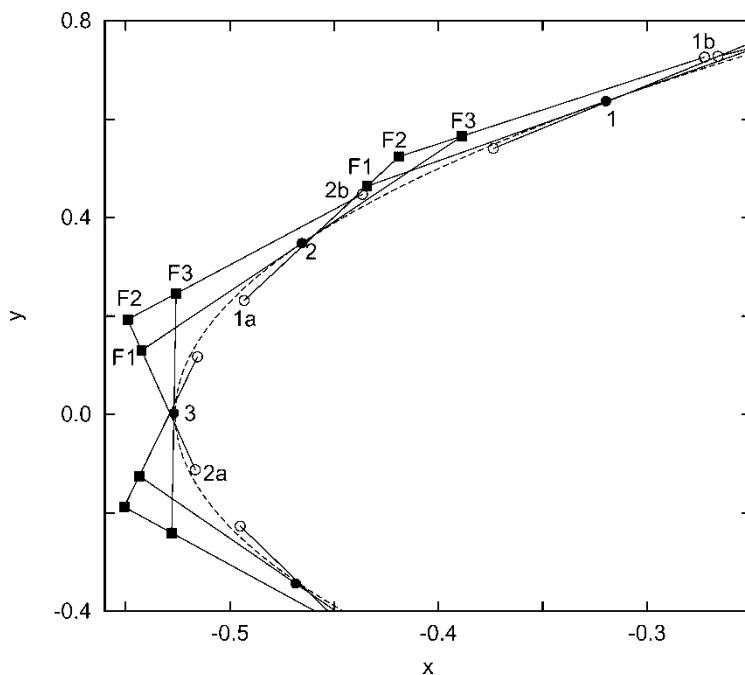


Figure A3. The intermediate positions and force evaluation points of the Forest-Ruth integrator. The dash curve is the exact orbit. See text for details.

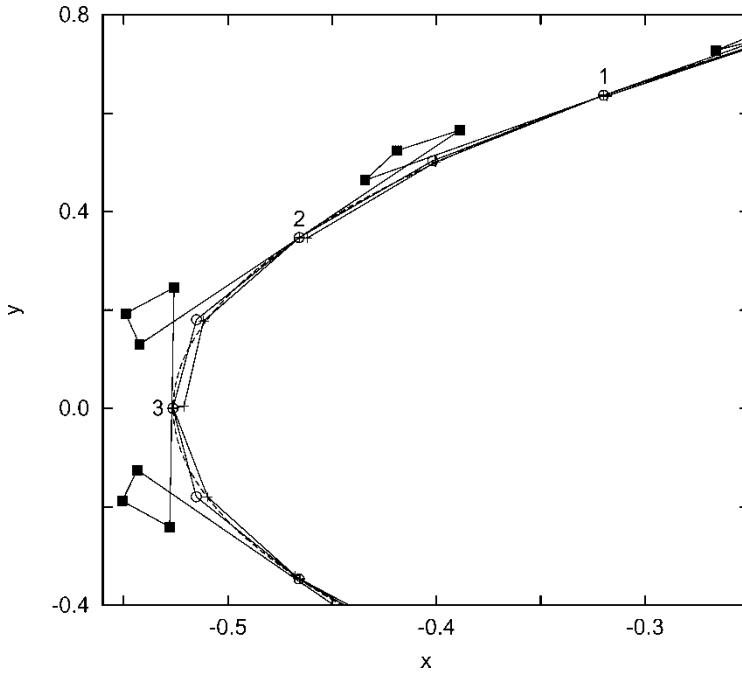


Figure A4. The force evaluation points of integrator FR (solid squares), forward integrator A (circles) and non-symplectic integrator RKN (plus signs).

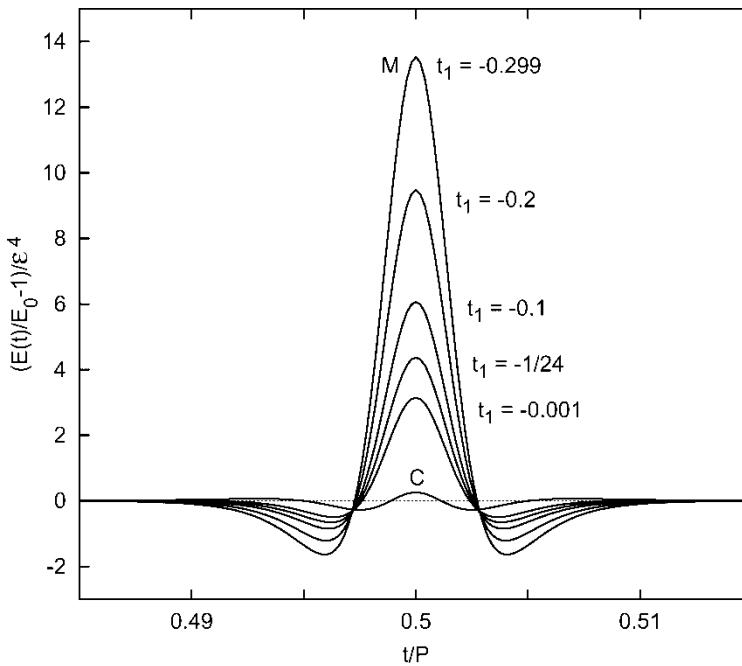


Figure A5. The fourth-order energy error coefficients of a family of non-forward integrators (27) with four force evaluations including that of McLachlan (M) as compared to that of forward integrator C. The parameter t_1 characterizes the size of the backward time step for updating the intermediate positions.

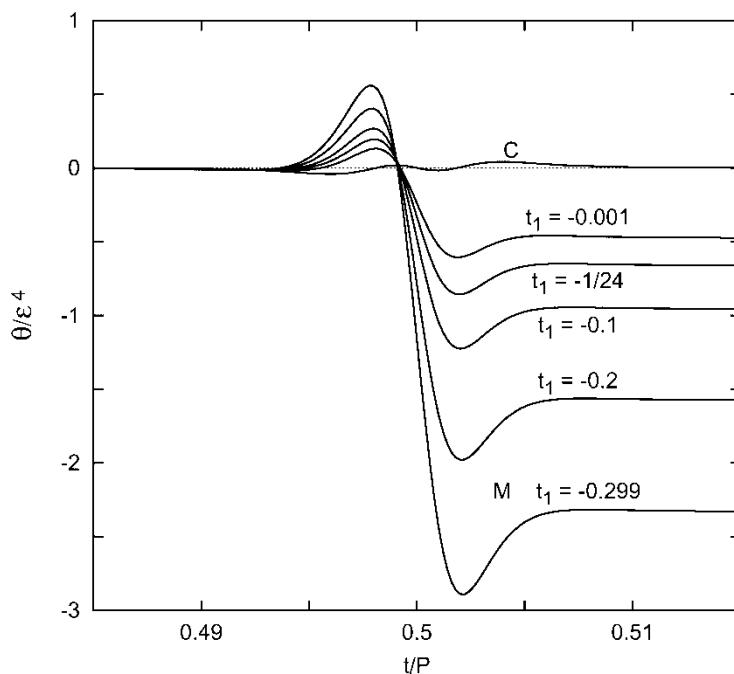


Figure A6. The fourth-order orbital precession error coefficient as measured by the rotation angle of the Laplace-Runge-Lenz vector for integrators described in figure A5.

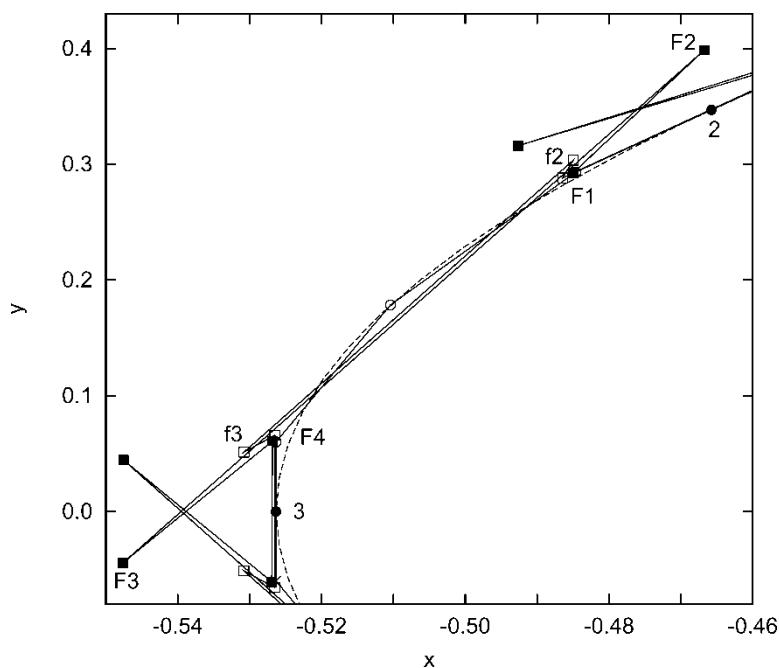


Figure A7. The force evaluation points of Mclachlan's integrator M (solid squares), reduced backward time step integrator with $t_1 = -1/24$ (hollow squares) and forward algorithm C (circles). The solid circles denote the starting position 2 and the final position 3 after one time step.

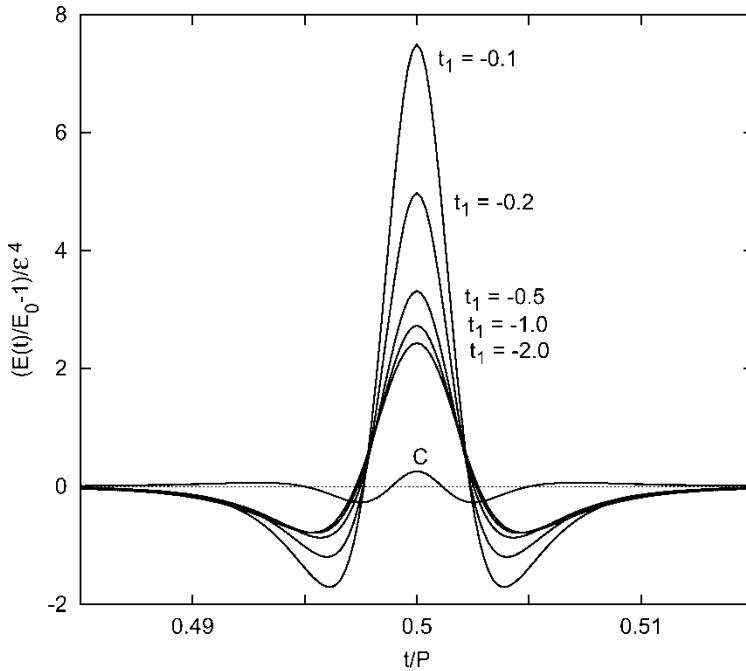


Figure A8. The fourth-order energy error coefficients of a family of non-forward integrators (33) with four force evaluations which updates the *momentum* first. Here, the more negative the parameter t_1 the smaller the negative time step size for updating the intermediate positions.

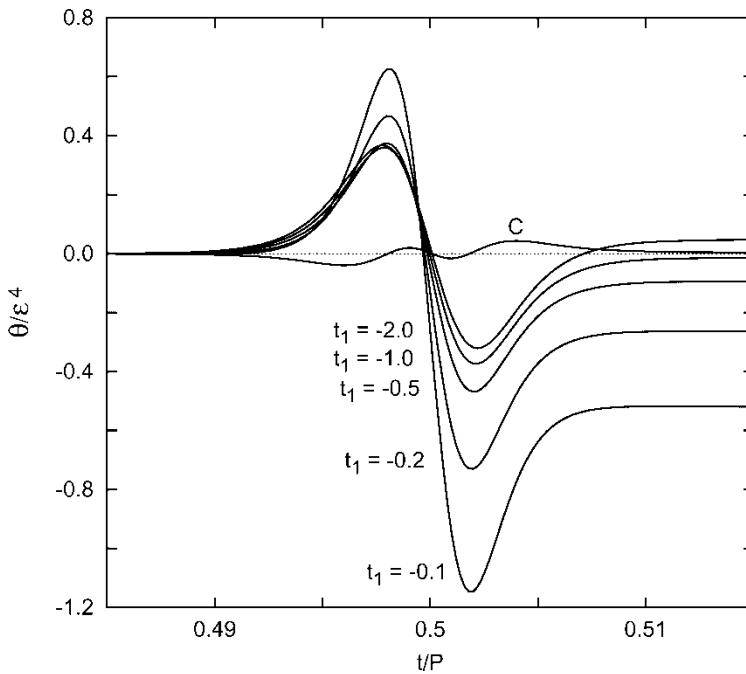


Figure A9. The fourth-order orbital precession error coefficient as measured by the rotation angle of the Laplace-Runge-Lenz vector for integrators described in figure A8. In this case, it is possible to fine tune t_1 so that the precession error is zero after each period.

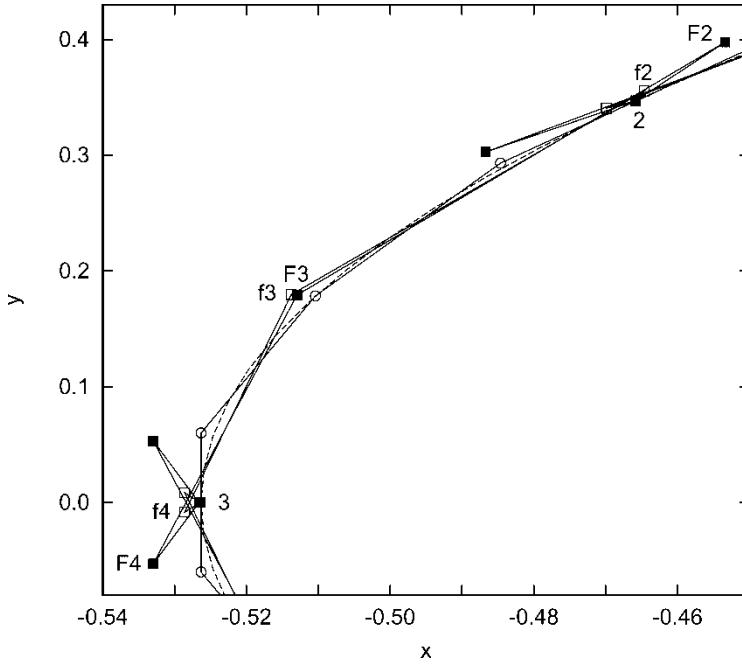


Figure A10. The force evaluation points of integrator (33) with $t_1 = -0.1$ (solid squares), the reduced backward time step integrator at $t_1 = -0.5$ (hollow squares) and the forward algorithm C (circles).

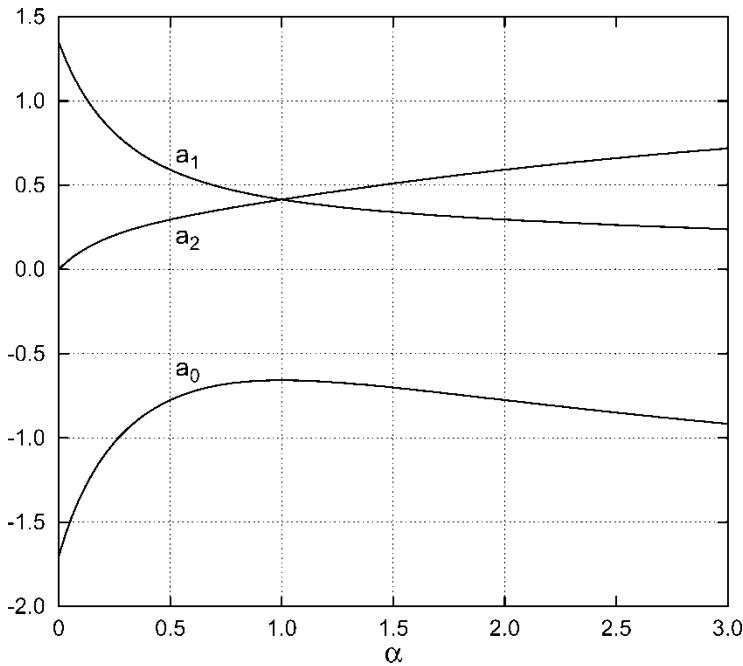


Figure A11. The coefficients of integrator (35) as a function of the free parameter α . The negative coefficient a_0 is least negative at $\alpha = 1$.

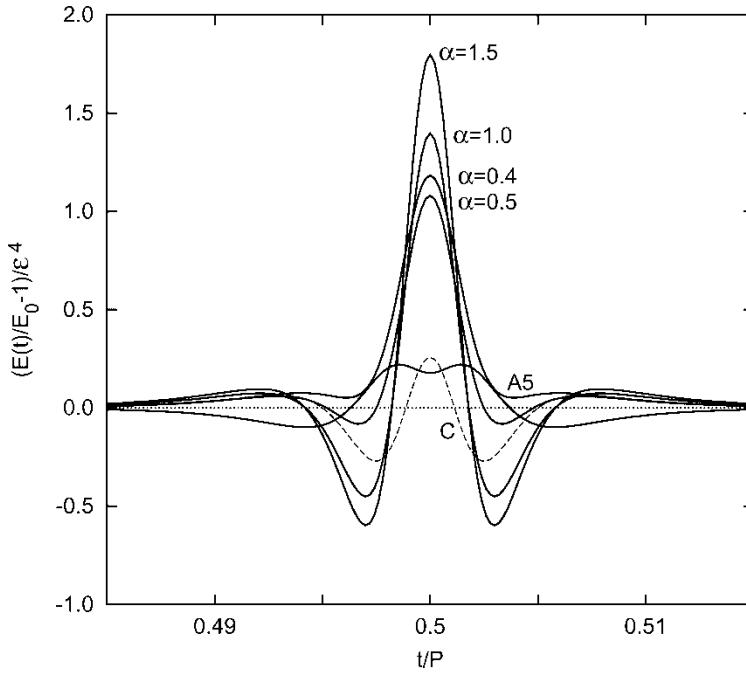


Figure A12. The fourth-order energy error coefficients of a family of non-forward integrators (35) with five force evaluations. These are compared to the five force forward integrator A5 and the four force forward integrator C.

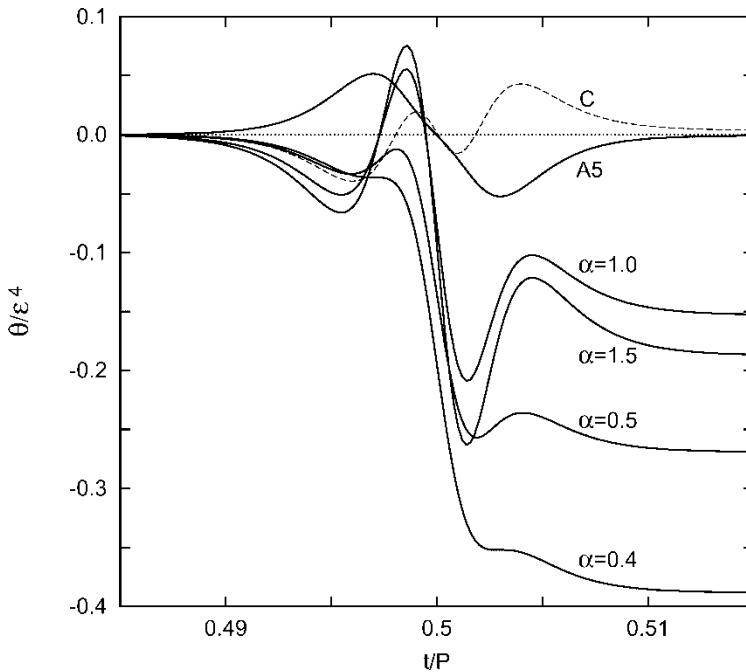


Figure A13. The fourth-order orbital precession error coefficient as measured by the rotation angle of the Laplace-Runge-Lenz vector for integrators described in figure A12.

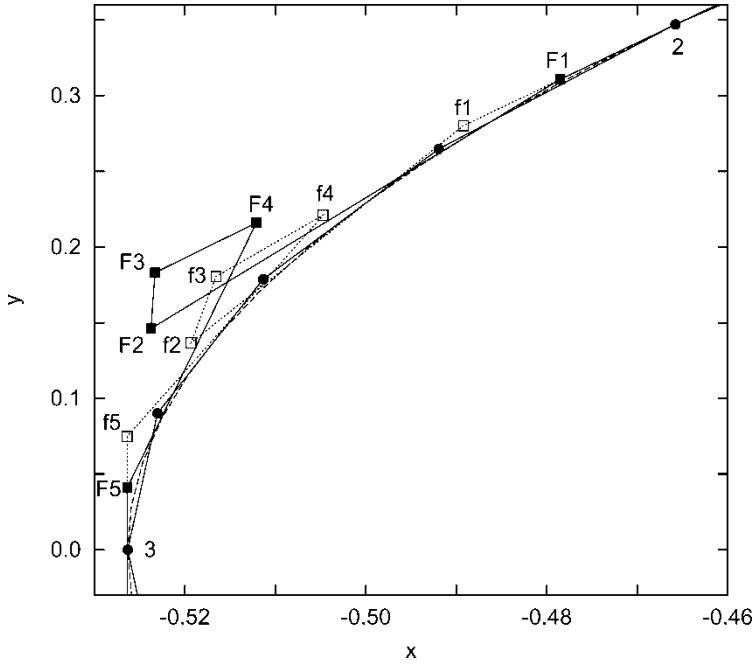


Figure A14. The force evaluation points of non-forward integrator (35) at $\alpha = 0.3$ (solid squares), with minimum backward time step at $\alpha = 1$ (hollow squares) and that of forward algorithm A5 (solid circles).

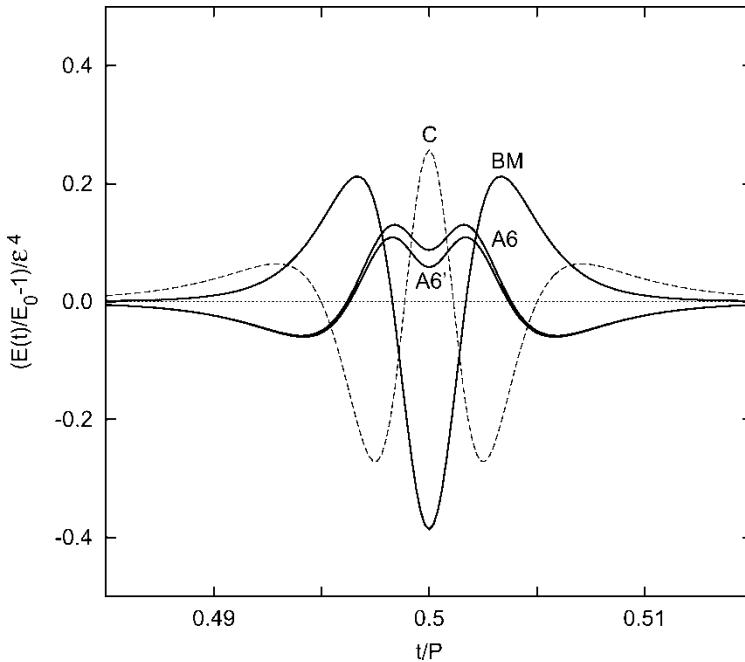


Figure A15. The fourth-order energy error coefficients of three integrators with six force evaluations. BM is Blanes and Moan's integrator [53]. A6 and A6' are forward integrators. A6' uses the extrapolated force gradient. Algorithm C is a four force forward integrator kept for comparison.

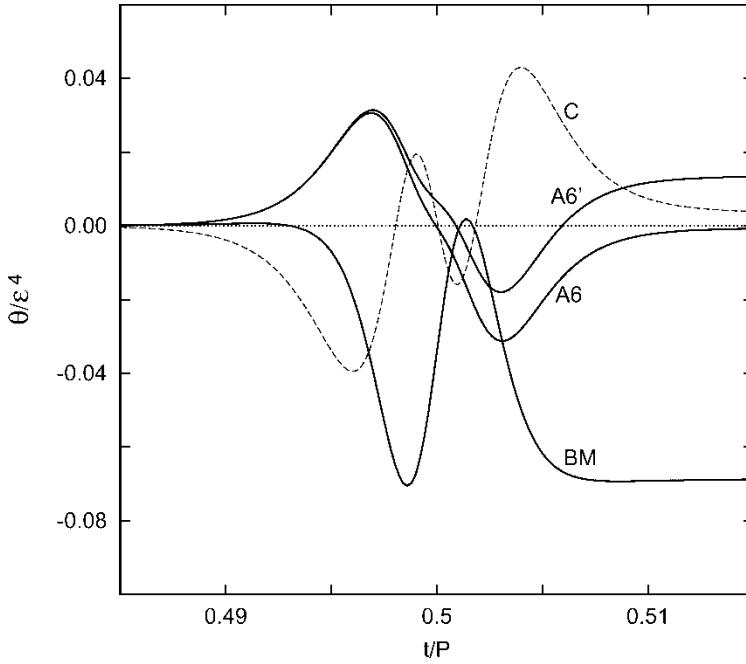


Figure A16. The fourth-order orbital precession error coefficient as measured by the rotation angle of the Laplace-Runge-Lenz vector for integrators described in figure A15. The extrapolated gradient integrator A6's precession error is much larger than that of A6, but still smaller than that of BM.

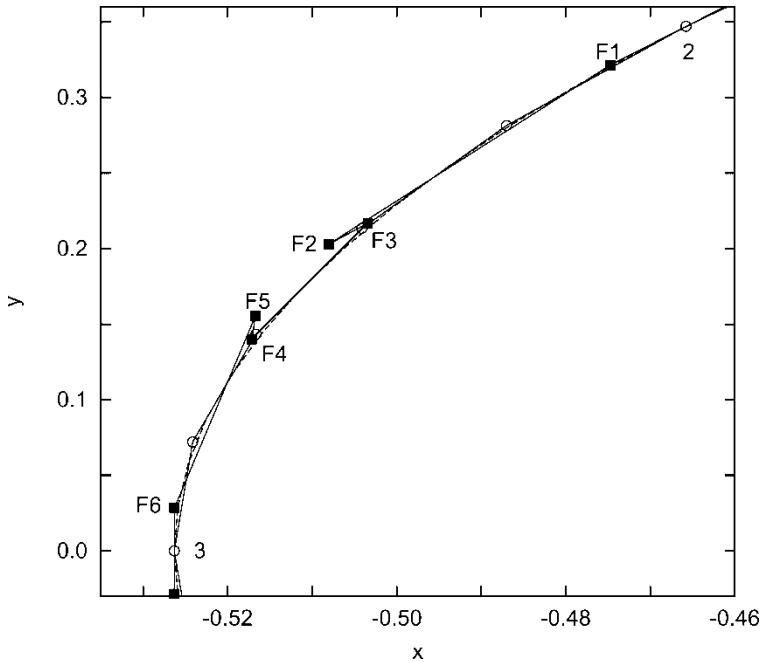


Figure A17. The force evaluation points of non-forward integrator BM (solid squares) and that of forward integrator A6 (circles).