Celestial Mechanics and Dynamical Astronomy (2005) 91:301–322
DOI 10.1007/s10569-004-4622-z

FORWARD SYMPLECTIC INTEGRATORS FOR SOLVING GRAVITATIONAL FEW-BODY PROBLEMS

SIU A. CHIN and C. R. CHEN
Center for Theoretical Physics, Department of Physics, Texas A&M University, College Station, TX 77843, e-mail: chin@physics.tamu.edu

(Received: 19 May 2004; revised: 18 August 2004; accepted: 23 September 2004)

Abstract. We introduce a class of fourth order symplectic algorithms that are ideal for doing long time integration of gravitational few-body problems. These algorithms have only positive time steps, but require computing the force gradient in addition to the force. We demonstrate the efficiency of these Forward Symplectic Integrators by solving the circular restricted three-body problem in the space-fixed frame where the force on the third body is explicitly time-dependent. These algorithms can achieve accuracy of Runge–Kutta, conventional negative time step symplectic and corrector symplectic algorithms at step sizes five to ten times as large.

Key words: symplectic integrators, positive time steps, three-body orbits, long time simulations

1. Introduction

Symplectic Integrators (SI) (Yoshida 1993; Channell and Neri 1996; McLachlan et al. 2002) conserve all Poincaré invariants when integrating classical trajectories. For periodic orbits, their energy errors are bounded and periodic, in contrast to Runge-Kutta type algorithms whose energy error increases linearly with integration time (Gladman et al. 1991; Kinoshita et al. 1991). Symplectic algorithms are therefore better long time integrators of astrophysical many-body problems (Wisdom and Holman 1991) and have been studied extensively in the literature (Candy and Rozmus 1991; Yoshida 1993; McLachlan 1995a; Koseleff 1996; Blanes et al. 1999; McLachlan et al. 2002). However, current higher order SI seem to suffer two disadvantages. First, the number of force evaluations required by these algorithms proliferates much more rapidly than non-symplectic algorithms. For example, a sixth order Runge–Kutta–Nöstrom algorithm (Albrecht 1955) requires only five force evaluations; a sixth order symplectic algorithm with negative intermediate time steps requires at least seven force evaluations (Yoshida 1990). Second, they are stable only for very small time-step sizes. This may also be due to the use of negative time steps for cancelling lower order errors. In this work, we call attention to a class of fourth order Forward Symplectic

Integrators (FSI), which have only positive time steps (Chin 1997; Chin and Kidwell 2000; Chin and Chen 2002). (We have referred to these as gradient symplectic algorithms previously.) The price for having only positive time steps is that one must compute the force gradient in addition to the force. This additional effort in solving gravitational few-body problems is very modest, but the resulting gain in numerical efficiency and stability is tremendous. We will show below that these FSI can achieve accuracy of existing symplectic integrators at step sizes five to ten times as large. These very powerful algorithms deserve to be considered by colleagues doing long time few-body simulations in various celestian mechanics contexts.

Despite the fact that Wisdom and Holman (1991) have questioned in 1991 why there are no FSI beyond second order, this class of algorithms has been unrecognized and overlooked until recently. The reason for this is simple; classical dynamics is time reversible, there is no compelling reason to require purely positive time steps. The demand for positive time steps only arises when one is solving *time-irreversible* equations, such the diffusion, Fokker–Planck, Navier–Stokes or the imaginary time Schrödinger equations. Based on FSI to be described, we have derived extremely efficient algorithms for solving and simulating the imaginary time Schrödinger equation (Auer et al. 2001; Forbert and Chin 2001b; Ciftja and Chin 2003; Chin 2004) and the Fokker–Planck/Langevin equation (Forbert and Chin 2001a). Our algorithms have also been used by others to solve quantum statistical and path-integral problems (Jang et al. 2001). However, once these FSI have been derived, they can be used to solve *both* time-reversible and time-irreversible dynamical problems. For time-irreversible systems, they are the only decomposition algorithms possible. However, even for time-reversible problems, they have been shown to be very efficient in solving both classical (Chin 1997; Chin and Kidwell 2000) and quantum (Chin and Chen 2001, 2002) dynamical problems. In this work, we demonstrate that these FSIs are equally effective to solve gravitational few-body problems with an explicitly time-dependent force field.

In the next Section, we will describe in detail how this class of algorithms fits into the overall picture of symplectic integrator development. We will show that our FSI traces its lineage to Ruth's seldom used third order algorithm (Ruth 1983). In Section 3, we describe our fourth order FSI and the one-parameter family of integrators. In Section 4, we compare various algorithms by computing an intricate, closed orbit of a circular restricted three-body problem. Physical three-body problems, such as the Sun–Earth–Moon, or Sun–Jupiter–Saturn, are too ''tame'' for discriminating the merit of these powerful algorithms. We therefore choose to solve a restricted three-body problem with many close encounters. We summarize our conclusions and suggest some future direction of research in Section 5.

2. Relation to Existing Symplectic Integrators

To give context to our work, we will review the development of symplectic integrators [see the excellent review by Yoshida (1993)] in the general context of solving any evolution equation of the form

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

where T and V are non-commuting operators. Classically, the evolution of a dynamical variable $w(q_i, p_i)$ is given by the Poisson bracket (sum over repeated indices),

$$\frac{\mathrm{d}}{\mathrm{d}t} w(q_i, p_i) = \{w, H\} \equiv \left(\frac{\partial w}{\partial q_i} \frac{\partial H}{\partial p_i} - \frac{\partial w}{\partial p_i} \frac{\partial H}{\partial q_i} \right). \tag{2}$$

For a Hamiltonian of the form,

$$H(p, q) = \frac{1}{2} p_i p_i + v(q_i), \tag{3}$$

Equation (2) is of the form (1) with T and V given by

$$T = p_i \frac{\partial}{\partial q_i}, \qquad V = F_i \frac{\partial}{\partial p_i}, \tag{4}$$

where $F_i = -\partial v / \partial q_i$. In this case, the constituent evolution operators $\mathrm{e}^{\epsilon T}$ and $\mathrm{e}^{\epsilon V}$ displace q_i and p_i forward in time via

$$q_i \to q_i + \epsilon p_i \qquad \text{and} \qquad p_i \to p_i + \epsilon F_i. \tag{5}$$

In general, the evolution Equation (1) can be solved iteratively via

$$w(t + \epsilon) = \mathrm{e}^{\epsilon(T+V)} w(t), \tag{6}$$

provided that one has a suitable approximation for the short time evolution operator $\mathrm{e}^{\epsilon(T+V)}$. If $\mathrm{e}^{\epsilon T}$ and $\mathrm{e}^{\epsilon V}$ can be exactly implemented individually, then $\mathrm{e}^{\epsilon(T+V)}$ can be decomposed to arbitrarily high order in the form

$$\mathrm{e}^{\epsilon(T+V)} = \prod_i \mathrm{e}^{a_i \epsilon T} \mathrm{e}^{b_i \epsilon V}. \tag{7}$$

For example, from the second order factorization

$$\mathrm{e}^{\epsilon(T+V)} \approx \mathrm{e}^{\frac{1}{2}\epsilon T} \mathrm{e}^{\epsilon V} \mathrm{e}^{\frac{1}{2}\epsilon T} = \mathscr{T}^{(2)}, \tag{8}$$

one can construct a fourth order approximation via

$$\mathscr{T}^{(4)}_{FR}(\epsilon) = \mathscr{T}^{(2)}(\widetilde{\epsilon}) \mathscr{T}^{(2)}(-s\widetilde{\epsilon}) \mathscr{T}^{(2)}(\widetilde{\epsilon}) \tag{9}$$

where $s = 2^{1/3}$ is chosen to cancel the third order error in $\mathscr{T}^{(2)}$ and $\widetilde{\epsilon} = \epsilon/(2-s)$ rescales the sum of forward–backward–forward time steps back to ϵ. This fourth order Forest–Ruth (FR) scheme (Forest and Ruth 1990), has been independently derived in diverse disciplines (Campostrini and Rossi 1990; Candy and Rozmus 1991). The above construction, while widely known through the work of Yoshida (1990), was first published by Creutz and Gocksch (1989).

The middle time step in (9) is negative. This is not accidental. Sheng (1989) and Suzuki (1991) have independently proved that, *beyond second order, any factorization of the form (7) must contain some negative coefficients in the set* $\{a_i, b_i\}$. Goldman and Kaper (1996) later proved that any factorization of the form (7) must contain at least one negative coefficients for *both* operators. This means that for any evolution equations containing an irreversible operator, such as the diffusion kernel $T = \frac{1}{2}\nabla^2$, no algorithms of the form (7) is possible beyond second order. This is because $\langle \mathbf{r}'|e^{a_i \epsilon T}|\mathbf{r}\rangle \propto e^{-(\mathbf{r}'-\mathbf{r})^2/(2a_i\epsilon)}$ is the diffusion kernel only if a_i is positive. If a_i were negative, then the kernel would be unbound and unnormalizable, reflecting the fact that diffusion is a time-irreversible process. Positive coefficients are therefore absolutely essential for solving any evolution equation having an irreversible component. We have shown, and will further demonstrate below, that even for time-reversible systems such quantum (Chin and Chen 2001, 2002) or classical dynamics (Chin 1997; Chin and Kidwell 2000; Omelyan et al. 2002, 2003), purely positive coefficients give rises to very efficient algorithms with excellent convergence properties.

While Sheng's proof is slightly more general, Suzuki's proof provides insight into how to circumvent this negative coefficient problem. The essence of Suzuki's proof is to note that, for example, one has the following operator representation of the velocity form of the Verlet algorithm,

$$e^{\frac{1}{2}\epsilon V} e^{\epsilon T} e^{\frac{1}{2}\epsilon V} = e^{\epsilon H'}, \tag{10}$$

with

$$H' = T + V - \frac{1}{12}\epsilon^2 [T,[V,T]] + \frac{1}{24}\epsilon^2 [V,[T,V]] + O(\epsilon^4). \tag{11}$$

In order to obtain a fourth order algorithm, one must eliminate the two double commutator error terms above. With purely positive coefficients a_i and b_i, one can eliminate either one, but not both. Thus to obtain a fourth order factorization with only positive coefficients, one must *keep* one of the two double commutators. With T and V as defined by (4), the commutator

$$[V,[T,V]] = 2F_j \frac{\partial F_i}{\partial q_j}\frac{\partial}{\partial p_i} = \nabla_i(|\mathbf{F}|^2)\frac{\partial}{\partial p_i}, \tag{12}$$

simply give rises to a new force defined by the gradient of the square of the force modulus. This commutator can therefore be kept. The use of this force gradient is not new. Ruth (1983) in his pioneering paper has derived a third order derivative algorithm on the basis of canonical transformations. However, this algorithm was universally ignored (Channell and Neri 1996) with no follow-up discussion. In 1997 one of us (Chin 1997) noted that Ruth's algorithm actually corresponds to an asymmetric operator decomposition and that his force derivative is precisely the force gradient required by $[V,[T,V]]$. The importance of Suzuki's proof is that it draws a clear connection between the demands for positive time steps and the necessity of keeping higher order commutators, such as $[V,[T,V]]$.

If one were to keep $[V,[T,V]]$, then there are two distinct ways of eliminating the other double commutator $[T,[V,T]]$. Despite the lack of an operator formalism, Rowlands (1991) also noted that the velocity form of the Verlet algorithm has error terms of the form (11). He proposes to get rid of the $[T,[V,T]]$ term by an explicit transformation and to view the algorithm as fourth order for a Hamiltonian with a modified potential

$$V' = V + \frac{1}{24}\epsilon^2[V,[T,V]]. \tag{13}$$

The reinterpretation of the potential is easy, the tricky part here is that since one has performed a transformation to eliminate $[T,[V,T]]$, one must transform back correctly to preserve fourth order accuracy for the original variables. This transformation is often more complicated than the original algorithm. This way of eliminating $[T,[V,T]]$ is tantamount to applying operators

$$\mathrm{e}^{C(\epsilon)}\mathrm{e}^{\frac{1}{2}\epsilon V}\mathrm{e}^{\epsilon T}\mathrm{e}^{\frac{1}{2}\epsilon V}\mathrm{e}^{-C(\epsilon)} \tag{14}$$

where $\mathrm{e}^{-C(\epsilon)}$ and $\mathrm{e}^{C(\epsilon)}$ are the initial and final transformations respectively. They can be further decomposed into the product form,

$$\mathrm{e}^{C(\epsilon)} = \prod_i \mathrm{e}^{c_i\epsilon T}\mathrm{e}^{d_i\epsilon V}.$$

This class of "process" (Lopez-Marcos et al. 1996, 1997; Blanes et al. 1999) or "corrector" (Wisdom et al. 1996; McLachlan 1996) symplectic algorithms has the distinct advantage that when one iterates the algorithm, only the kernel algorithm needs to be iterated. If there is no need to keep track of intermediate results, then a fourth order algorithm is possible with only a single evaluation of the modified force (Lopez-Marcos et al. 1996, 1997; Wisdom et al. 1996; Blanes et al. 1999). However, by its very construction, either $\mathrm{e}^{-C(\epsilon)}$ or $\mathrm{e}^{C(\epsilon)}$ must contain negative time steps, and cannot be applied to equations with an irreversible operator, except when doing quantum

statistical *trace* calculations (Chin 2004). Process algorithms also uses a modified potential, but only in the form (13) and only in the context of a second order kernel algorithm (Blanes et al. 1999). Lopez-Marcos et al. (1997) have used a two-fold Rowlands kernel that could have been fourth order, but the correct parameter values for that were never considered.

In the operator formalism, a more direct way of eliminating $[T,[V,T]]$ is to add more operators symmetrically. In this approach, the elimination of $[T,[V,T]]$ is built-in, with no need of an extrinsic transformation. McLachlan (1995b) has done this for algorithms 4A and 4B (described below) in the context of a slightly perturbed Hamiltonian system. But his algorithms were not truly fourth order because he did not *keep* the commutator $[V,[T,V]]$. These pseudo-higher order algorithms (McLachlan 1995b; Chambers and Murison 2000; Laskar and Robutel 2001), which only require a single error commutator to vanish at each order, are much simpler than regular higher order algorithms, which require *all* error commutators to vanish at each order. There is also a widely cited reference by Koseleff (1993) that purported to contain a fourth order scheme using modified potentials. Koseleff's algorithm would have been algorithm 4A discussed below, unfortunately, his coefficient for the commutator $[V,[T,V]]$ is *incorrect*. It should be 1/72 rather than 1/24.

3. Fourth Order Forward Symplectic Integrators

Suzuki, on the basis of McLachlan's work (McLachlan 1995b), retained the commutator $[T,[V,T]]$ and wrote down two fourth order factorization schemes 4A and 4B with purely positively coefficients (Suzuki 1996). He did not implement them classically or quantum mechanically, nor demonstrated their usefulness. One of us (Chin 1997) derived schemes 4A and 4B by elementary means independent of McLachlan's work, and have explicitly implemented them classically and demonstrated their effectiveness in solving the Kepler problem. Moreover, a new algorithm 4C (Chin 1997) was found, which is the direct descendant of Ruth's third order algorithm (Ruth 1983). In all cases tested, algorithm 4C has much smaller error coefficients than 4A and 4B. Prior to Chin's work, we are not aware of any implementation of fourth order FSI for solving any problem.

Since the commutator and the gradient force term occur frequently in the following, we will define

$$U(t) \equiv [V(t),[T,V(t)]] \quad \text{and} \quad \mathbf{G}_i(t) \equiv \nabla_i(|\mathbf{F}(t)|^2) \tag{15}$$

to simplify notation. The time-dependent form of the four FSI derived in Chin and Chen (2002) are:

$$\mathscr{T}_A^{(4)}(\epsilon) \equiv e^{\frac{1}{6}\epsilon V(t+\epsilon)} e^{\frac{1}{2}\epsilon T} e^{\frac{2}{3}\epsilon \widetilde{V}(t+\epsilon/2)} e^{\frac{1}{2}\epsilon T} e^{\frac{1}{6}\epsilon V(t)}, \tag{16}$$

with $\widetilde{V}$ defined by

$$\widetilde{V}(t) = V(t) + \frac{1}{48}\epsilon^2 U(t), \tag{17}$$

corresponding to a modified force

$$\widetilde{\mathbf{F}}(t) = \mathbf{F}(t) + \frac{1}{48}\epsilon^2 \mathbf{G}(t). \tag{18}$$

Given $\mathbf{p}_0 = \mathbf{p}(t)$ and $\mathbf{q_0} = \mathbf{q}(t)$, transcribing each operator in (16) according to (5) yields the following explicit algorithm 4A for advancing the system forward from t to $t + \epsilon$,

$$\begin{aligned}
\mathbf{p}_1 &= \mathbf{p}_0 + \frac{1}{6}\epsilon \mathbf{F}(\mathbf{q}_0, t) \\
\mathbf{q}_1 &= \mathbf{q}_0 + \frac{1}{2}\epsilon \mathbf{p}_1 \\
\mathbf{p}_2 &= \mathbf{p}_1 + \frac{2}{3}\epsilon \widetilde{\mathbf{F}}(\mathbf{q}_1, t + \epsilon/2) \\
\mathbf{q}_2 &= \mathbf{q}_1 + \frac{1}{2}\epsilon \mathbf{p}_2 \\
\mathbf{p}_3 &= \mathbf{p}_2 + \frac{1}{6}\epsilon \mathbf{F}(\mathbf{q}_2, t + \epsilon).
\end{aligned} \tag{19}$$

The last $\mathbf{p}$ and $\mathbf{q}$ are the updated variables, *i.e.*, $\mathbf{q}(t+\epsilon) = \mathbf{q}_2$ and $\mathbf{p}(t+\epsilon) = \mathbf{p}_3$. As noted earlier (Chin 1997), algorithm 4A only requires two evaluations of the force (the last evaluation can be re-used) and one evaluation of the force gradient. Note that forces are to be evaluated at an intermediate time equals the sum of time steps of all the preceding T operators (Suzuki 1993; Chin and Chen 2002).

Similarly algorithm 4B in operator form is

$$\mathscr{T}_B^{(4)}(\epsilon) \equiv e^{t_2\epsilon T} e^{\frac{1}{2}\epsilon \bar{V}(t+a_2\epsilon)} e^{t_1\epsilon T} e^{\frac{1}{2}\epsilon \bar{V}(t+a_1\epsilon)} e^{t_0\epsilon T}, \tag{20}$$

where

$$t_0 = t_2 = \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right), \quad t_1 = \frac{1}{\sqrt{3}}, \quad a_1 = \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right), \quad a_2 = \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\right), \tag{21}$$

and with $\bar{V}$ given by

$$\bar{V}(t) = V(t) + c_0\epsilon^2 U(t). \tag{22}$$

This is just a modified force with a different coefficient $c_0 = \frac{1}{24}(2-\sqrt{3})$. The transcription to an explicit algorithm is the same as 4A and will not be repeated. Algorithm 4B requires two evaluations of the force and two evaluation of the force gradient. After we discovered the one-parameter family of algorithms discussed below, we realized that one can improve 4B by eliminating one evaluation of the gradient force. This is done by concentrating the commutator term at the center of algorithm:

$$\mathscr{T}^{(4)}_{B'}(\epsilon) \equiv e^{t_2\epsilon T} e^{\frac{1}{2}\epsilon V(t+a_2\epsilon)} e^{\frac{1}{2}t_1\epsilon T} e^{c_0\epsilon^3 U(t+\epsilon/2)} e^{\frac{1}{2}t_1\epsilon T} e^{\frac{1}{2}\epsilon V(t+a_1\epsilon)} e^{t_0\epsilon T}. \tag{23}$$

Similar to algorithm 4A, 4B′ only requires two force and one gradient evaluations for every update.

One of us has shown (Chin 1997) that Ruth's original third order algorithm (Ruth 1983) simply corresponds to the following asymmetric third order decomposition:

$$e^{\epsilon(T+V)} = e^{\epsilon\frac{1}{3}T} e^{\epsilon\frac{3}{4}V} e^{\epsilon\frac{2}{3}T} e^{\epsilon\frac{1}{4}V'}, \tag{24}$$

with $V' = V + \frac{1}{12}\epsilon^2[V,[T,V]]$. The power of the operator approach is that one immediately realizes that the third order error in (24) can be eliminated by symmetrization, since a symmetric decomposition can only have errors of even order in ϵ. To symmetrize, take two third order algorithms (24) at half the time step, flip one over and concatenate. Algorithms 4C and 4D correspond to the two possible ways of concatenation:

$$\mathscr{T}^{(4)}_{C}(\epsilon) \equiv e^{\frac{1}{6}\epsilon T} e^{\frac{3}{8}\epsilon V(t+5\epsilon/6)} e^{\frac{1}{3}\epsilon T} e^{\frac{1}{4}\epsilon \widetilde{V}(t+\epsilon/2)} e^{\frac{1}{3}\epsilon T} e^{\frac{3}{8}\epsilon V(t+\epsilon/6)} e^{\frac{1}{6}\epsilon T}, \tag{25}$$

$$\mathscr{T}^{(4)}_{D}(\epsilon) \equiv e^{\frac{1}{8}\epsilon \widetilde{V}(t+\epsilon)} e^{\frac{1}{3}\epsilon T} e^{\frac{3}{8}\epsilon V(t+2\epsilon/3)} e^{\frac{1}{3}\epsilon T} e^{\frac{3}{8}\epsilon V(t+\epsilon/3)} e^{\frac{1}{3}\epsilon T} e^{\frac{1}{8}\epsilon \widetilde{V}(t)}, \tag{26}$$

where $\widetilde{V}(t)$ is the same modified force (17) used in algorithm 4A.

Algorithms 4A, 4C and 4B′ are special cases of a one-parameter family of algorithms

$$\mathscr{T}^{(4)}_{ACB'}(\epsilon) \equiv e^{t_3\epsilon T} e^{v_3\epsilon V(t+a_3\epsilon)} e^{t_2\epsilon T} e^{\epsilon W(t+a_2\epsilon)} e^{t_1\epsilon T} e^{v_1\epsilon V(t+a_1\epsilon)} e^{t_0\epsilon T} \tag{27}$$

parameterized by t_0. Given t_0, the rest of the coefficients are:

$$t_1 = t_2 = \frac{1}{2} - t_0, \quad t_3 = t_0, \quad v_1 = v_3 = \frac{1}{6}\frac{1}{(1-2t_0)^2}, \quad v_2 = 1 - (v_1 + v_3), \tag{28}$$

$a_1 = t_0$, $a_2 = 1/2$, $a_3 = 1 - t_0$, with $W(t)$ defined by

$$W(t) = v_2 V(t) + u_0\epsilon^2 U(t), \tag{29}$$

and

$$u_0 = \frac{1}{12}\left[1 - \frac{1}{1-2t_0} + \frac{1}{6(1-2t_0)^3}\right]. \tag{30}$$

All coefficients are positive for t_0 in the range $[0, \frac{1}{2}(1-\frac{1}{\sqrt{3}}) \approx 0.21]$. At $t_0 = 0$, one has algorithm 4A. At $t_0 = 1/6 \approx 0.167$, one has algorithm 4C. At the upper limit of $t_0 = \frac{1}{2}(1-\frac{1}{\sqrt{3}})$, $v_2 = 0$, and one recovers algorithm 4B′. One can therefore change continuously from algorithm 4A to 4C to 4B′ by varying the parameter t_0. This is very useful in cases where any two of the above algorithms have convergence errors of opposite signs. By varying t_0, one can set that error to zero with no additional computational effort.

There is also a one parameter family of algorithms connecting algorithms 4B, a variant of 4D and 4A (Chin and Chen 2002). This family generally requires four force evaluations plus the force gradient. We will not consider it further in this work.

4. The Circular Restricted Three-Body Problem

We compare the efficiency of these FSI by solving the planar circular restricted 3-body problem defined by

$$\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2} = \mathbf{a}(\mathbf{r},t) \equiv -\frac{1}{2}\Big[\mathbf{a}_1(\mathbf{r},t) + \mathbf{a}_2(\mathbf{r},t)\Big] \tag{31}$$

where for $i = 1, 2$,

$$\mathbf{a}_i(\mathbf{r},t) = \frac{\mathbf{r} - \mathbf{r}_i(t)}{S_i^3} \quad \text{and} \quad S_i = |\mathbf{r} - \mathbf{r}_i(t)|.$$

The two attractive centers $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ orbit about the origin in circles with angular velocity $\omega = 1$:

$$\mathbf{r}_1(t) = -\frac{1}{2}(\cos(t), \sin(t)),$$

$$\mathbf{r}_2(t) = \frac{1}{2}(\cos(t), \sin(t)).$$

The gradient term is no more complicated than the force itself,

$$\nabla(|\mathbf{a}|^2) = -\frac{1}{2}\Big[C_1\mathbf{a}_1 + C_2\mathbf{a}_2\Big] \tag{32}$$

with

$$C_1 = 2S_1^{-3} + 3S_1\mathbf{a}_1 \cdot \mathbf{a}_2 - S_2^{-3}$$

$$C_2 = 2S_2^{-3} + 3S_2\mathbf{a}_1 \cdot \mathbf{a}_2 - S_1^{-3}$$

We solve the problem directly in the space-fixed frame in which the force, or the acceleration field as defined above, is explicitly time-dependent. The initial condition $\mathbf{r}_0 = (0, 0.0580752367)$ and $\mathbf{v}_0 = (0.489765446, 0)$ produce an intricate and lengthy "Chinese Coin" orbit, useful for testing algorithms. The actual period is 18π, but in the present case where the two attracting centers are indistinguishable, the orbit repeats after 9π. For this work we will consider the orbital period to be $P = 9\pi$.

In Figure 1 we compare the trajectory after three periods ($t = 27\pi$) using the Forest–Ruth algorithm and our 4B′ algorithm. In Figure 2 we compare the same trajectory using algorithm M and algorithm 4C. M is McLachlan's fourth order, four force-evaluation algorithm (McLachlan 1995a):

$$\mathscr{T}_M^{(4)}(\epsilon) \equiv \mathrm{e}^{t_5\epsilon T}\mathrm{e}^{v_1\epsilon V(t+a_4\epsilon)}\mathrm{e}^{t_4\epsilon T}\mathrm{e}^{v_2\epsilon V(t+a_3\epsilon)}\mathrm{e}^{t_3\epsilon T}\mathrm{e}^{v_2\epsilon V(t+a_2\epsilon)}\mathrm{e}^{t_2\epsilon T}\mathrm{e}^{v_1\epsilon V(t+a_1\epsilon)}\mathrm{e}^{t_1\epsilon T}, \tag{33}$$

where $v_1 = 6/11$, $v_2 = 1/2 - v_1$, $a_i = \sum_{j=1}^{i} t_j$ and

$$t_1 = t_5 = \frac{642 + \sqrt{471}}{3924}, \quad t_2 = t_4 = \frac{121}{3924}(12 - \sqrt{471}), \quad t_3 = 1 - 2(t_1 + t_2).$$

Algorithms RF and M are representative fourth order symplectic algorithms with negative intermediate time steps. Both RF and M are outperformed by 4B′, which only requires two force and one gradient evaluation. Newer negative time steps fourth order algorithms, as recommend by McLachlan et al. (2002), require more than four force evaluations, are also not competitive (Scuro and Chin 2004). We have also tested the standard four force-evaluation Runge–Kutta and the three force-evaluation Runge–Kutta–Nöstrom algorithm (Battin 1999). At the same step size, both Runge–Kutta type algorithms are unstable and cannot produce a bounded trajectory. Algorithm 4A and 4B are comparable to 4B′, with 4D besting even 4C because it uses one more force evaluation.

The accuracy of each algorithm can be quantitatively assessed by monitoring the Jacobi constant. The Jacobi constant is usually given in the co-rotating frame in which the two attractive centers are at rest. However, it is just as easy to transform the Jacobi constant back to the space-fixed frame where it has a simple interpretation. For our circular restrict three-body problem defined by (31), the Jacobi constant J is given by

$$J = \mathbf{v}^2 - \frac{1}{|\mathbf{r} - \mathbf{r}_1(t)|} - \frac{1}{|\mathbf{r} - \mathbf{r}_2(t)|} - 2\mathbf{r} \times \mathbf{v}, \tag{34}$$

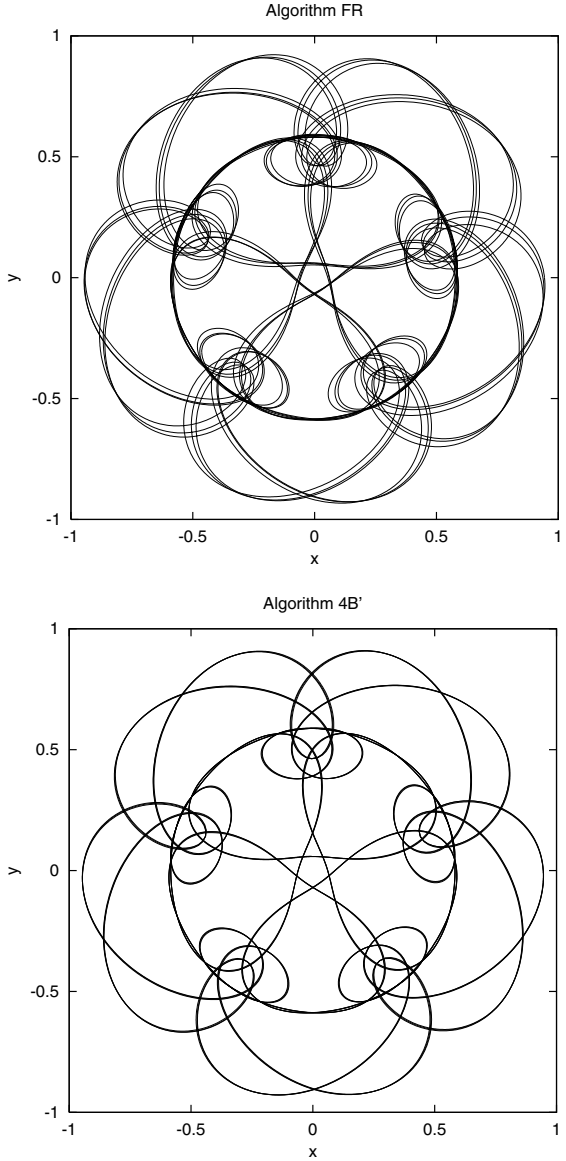

Figure 1. The trajectory of the third body in the space-fixed frame after three orbits in a circular restricted three-body problem. The time step used is very large, $\epsilon = 9\pi/5000 \approx 0.0057$. FR is the Forest–Ruth algorithm which uses three force evaluations per update. Algorithm 4B′, Equation (23), uses two force and one force gradient evaluations. At this large step size, both fourth order Runge–Kutta and the Runge–Kutta–Nostrom algorithms are unstable, producing only chaotic trajectories shooting off to infinity.

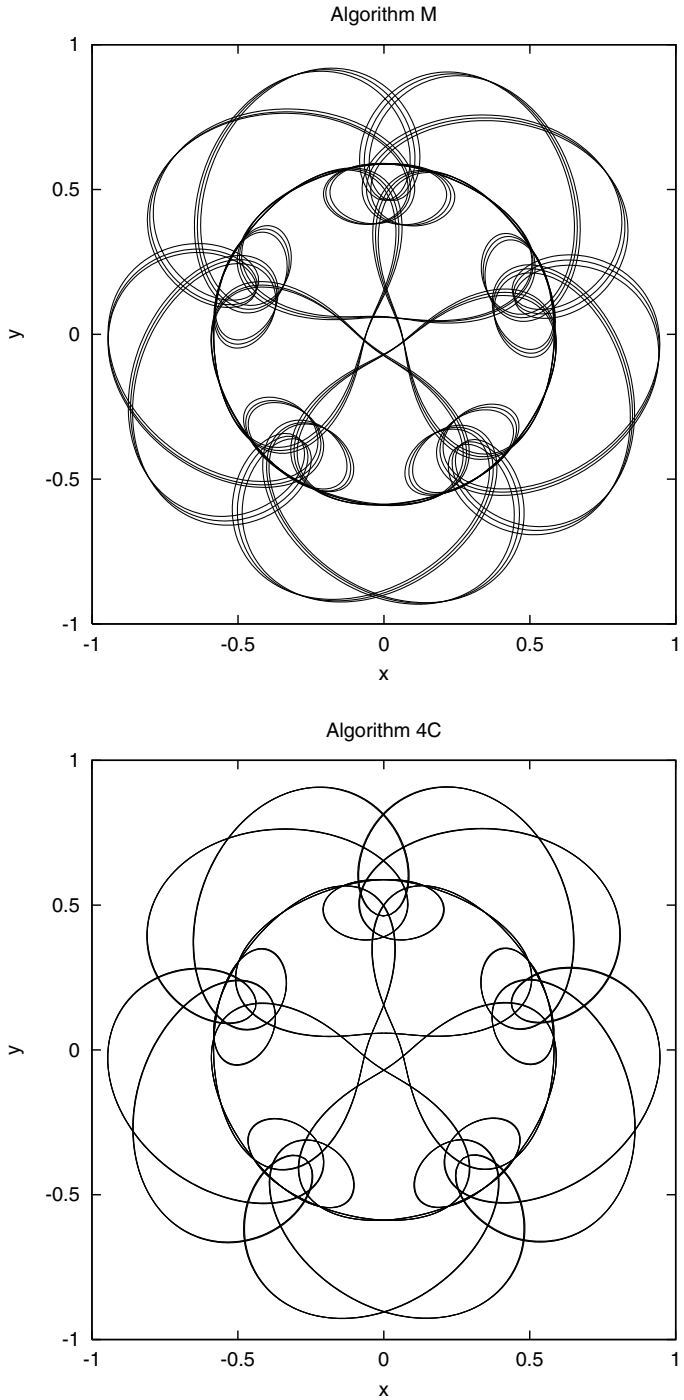

Figure 2. Same number of orbit and time step size as in Figure 1. M is McLachlan's fourth order algorithm which uses four force evaluations per update. Algorithm 4C, Equation (25), uses three force and one force gradient.

which is twice the difference between the energy and the angular momentum. For fourth order algorithms, we expect $J(t) = J_0 + \epsilon^4 J_4(t)$ and the step-size independent error coefficient $J_4(t)$ can be extract from

$$J_4(t) = \lim_{\epsilon \to 0} \frac{(J - J_0)}{\epsilon^4}.$$

As one decreases the step size, $J_4(t)$ converges to a fixed shape characteristic of the algorithm. In Figure 3 we show this coefficient function for the regular fourth order Runge–Kutta algorithm at a step size of $\epsilon = P/50000 \approx 0.00056$. In the course of one period, the third body has five close encounters with the two attractive centers, resulting in five error spikes at t/P = 1/10, 3/10, 5/10, 7/10, and 9/10. Between each close encounter there are also four minor encounters, resulting in barely discernable error blips. Figure 3 demonstrates the distinction between symplectic and non-symplectic algorithm. For symplectic algorithms, the error recovers back to zero after each close encounter and remain bounded after each period. For Runge–Kutta type algorithms, each encounter produces an irrecoverable error. The accumulating error then grows linearly in time without limit.

In Figure 4, we show the quantitative difference between forward and non-forward integrators by examining the detailed shape of the error spike at t/P = 1/10. In Figure 4a, the upper figure, we compare the error function $J_4(t)$ generated by algorithms FR, M and Cor with that of 4A. Cor is a processor or corrector algorithm of the form:

$$\mathscr{T}^{(4)}_{\mathrm{Cor}}(\epsilon) \equiv \mathrm{e}^{C(\epsilon)} \mathscr{T}^{(2)}_{2M}(\epsilon) \mathrm{e}^{-C(\epsilon)}, \tag{35}$$

where the kernel is a second order algorithm (2M)

$$\mathscr{T}^{(2)}_{2M}(\epsilon) \equiv \mathrm{e}^{\frac{1}{2}\epsilon T} \mathrm{e}^{\epsilon V'(t+\epsilon/2)} \mathrm{e}^{\frac{1}{2}\epsilon T} \tag{36}$$

with same modified force V' as defined by (13). This second order kernel has been used in most processor or corrector algorithms (Lopez-Marcos et al. 1996, 1997; Wisdom et al. 1996; Blanes et al. 1999). However, a fourth order corrector with analytical coefficients seems to have been overlooked in all of the above references. This corrector is (Chin 2004)

$$\mathrm{e}^{C(\epsilon)} = \mathrm{e}^{v_2 \epsilon V(t+[t_1+t_2]\epsilon)} \mathrm{e}^{t_2 \epsilon T} \mathrm{e}^{v_1 \epsilon V(t+t_1\epsilon)} \mathrm{e}^{t_1 \epsilon T}, \tag{37}$$

$$\mathrm{e}^{-C(\epsilon)} = \mathrm{e}^{-t_1 \epsilon T} \mathrm{e}^{-v_1 \epsilon V(t-t_2\epsilon)} \mathrm{e}^{-t_2 \epsilon T} \mathrm{e}^{-v_2 \epsilon V(t)}, \tag{38}$$

with

$$t_1 = \frac{1}{2\sqrt{3}}, \quad t_2 = -\frac{1}{2^{1/3}\sqrt{3}}, \quad v_1 = \frac{1}{2\sqrt{3}} - \frac{1}{2^{4/3}\sqrt{3}}, \quad v_2 = -\frac{1}{2^{4/3}\sqrt{3}}.$$

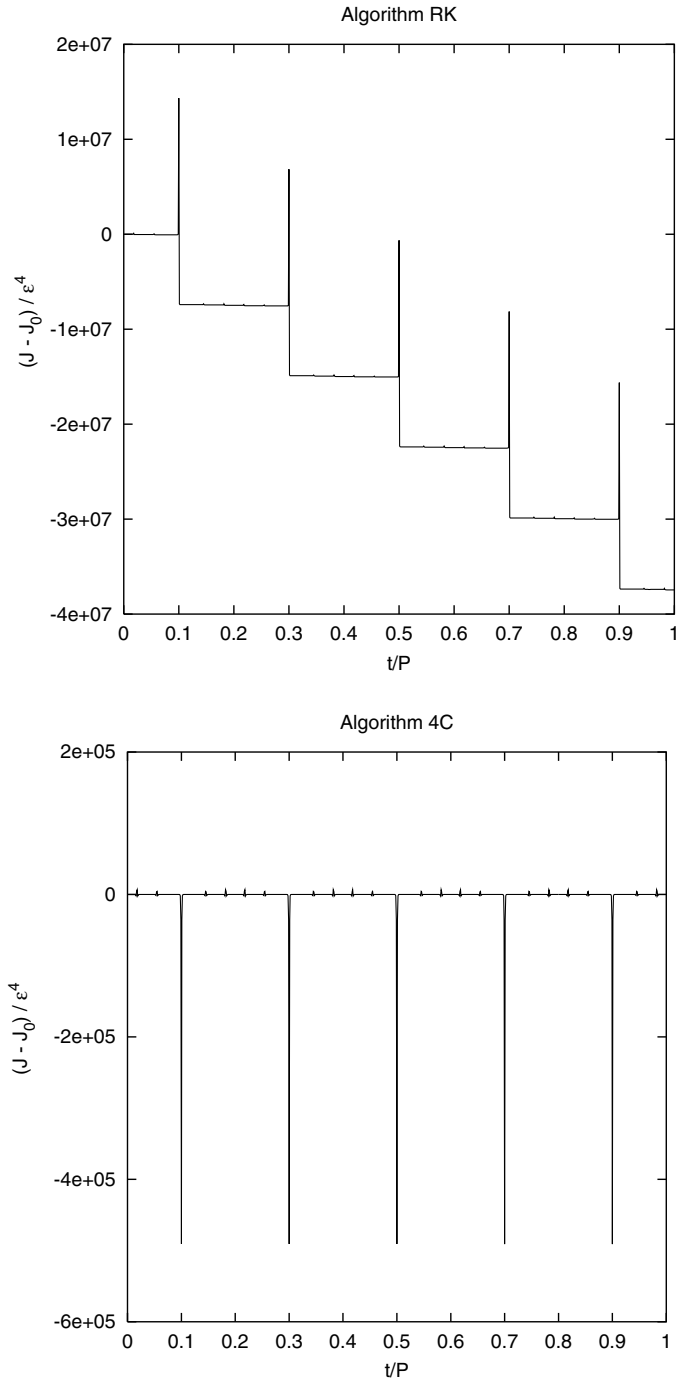

Figure 3. The error coefficient of the Jacobi constant as computed by the fourth order Runge–Kutta algorithm and the forward symplectic algorithm 4C. Note the relative scale, algorithm 4C's error coefficient is two orders of magnitude smaller.

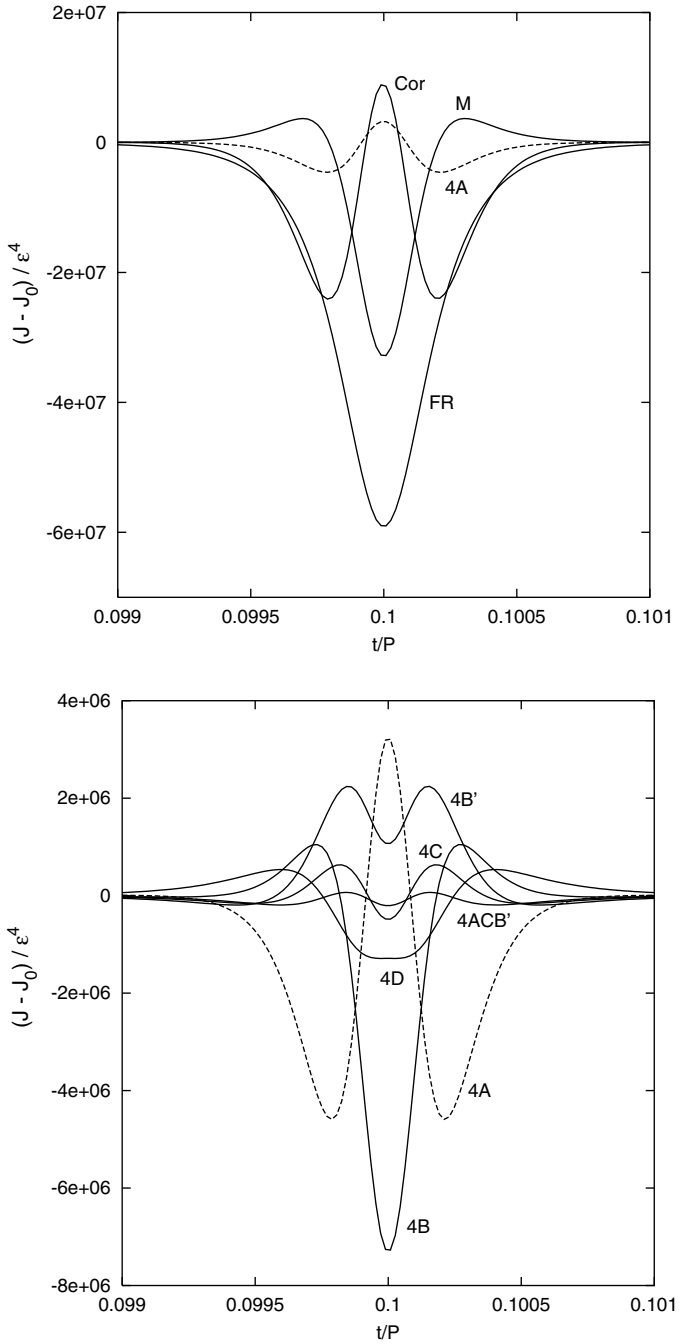

Figure 4. The error coefficient of the Jacobi constant at $t/P = 1/10$ due to non-forward symplectic algorithms FR, M and Cor (top) and forward symplectic algorithms (bottom). Algorithm 4A is drawn with dotted lines in both for comparison.

Note that at the end of $e^{-C(\epsilon)}$, time is to be set to $t-(t_1+t_2)\epsilon$. This is the fastest fourth order algorithm when no intermediate results are needed, requiring only one evaluation of the force and its gradient. (If output is needed after every update, as it is here, it is the slowest algorithm with five force and one gradient evaluations. Of course, this is not the way corrector algorithms are supposed to be used. The corrector should only be applied sparingly.) Its error function is a factor of two smaller than RF's, comparable to that of M, but an order of magnitude greater than that of 4A. In Figure 4b, the lower figure we compare all the FSI considered in this work. Algorithm 4B′, with one less force gradient evaluation, has only half the error of 4B. The one-parameter algorithm 4ACB′ with $t_0=0.138$ has only one-third the error of 4C. The error height of 4ACB′ is fully two orders of magnitude smaller than those of non-forward symplectic integrators. In Table I, we give the inverse error height of each algorithm, $1/|h_i|$, normalized to that of RF. Thus, algorithm 4ACB′'s error height is ≈300 times smaller than that of RF and ≈ 150 times smaller than that of algorithm M.

To further assess the convergence properties of these algorithms, we first examine the convergence of the third body's energy after one period as a function of the step size ϵ. The results are plotted in Figure 5. All can be very well fitted by single fourth order monomial $c_i\epsilon^4$ as shown by solid lines. One can basically read off the efficiency ranking of each algorithm from left to right following flatness of the convergence curve. The convergence of RK and RKN are similar, despite the fact that RK uses one more force evaluation. For this calculation, algorithm Cor is very efficient, requiring only one evaluation of the force and its gradient to best FR. In this case, the second order kernel algorithm 2M is fourth order even without the use of the cor-

TABLE I

The inverse error height in the Jacobi constant and the inverse fourth order error coefficient in term of RF's value.

	RK/ RKN	RF	Cor	M	4A	4B	4B′	4C	4D	4ACB′
$\lvert h_i\rvert^{-1}$	–	1	2.5	2	13	8	26	94	45	295
$\lvert c_i\rvert^{-1}$ (t = P)	0.1	1	2	4.9	12	23	28	2200	2300	140
$\lvert c_i\rvert^{-1}$ (t = P/2)	0.3	1	1.1	0.9	4.3	4.3	∼250	46	21	∼2000

For example, algorithm 4ACB′'s maximum error in the Jacobi constant is 295 times smaller than that of FR's and 118 times smaller than that of M, McLachlan's algorithm. After one period, algorithm 4C's energy error coefficient is 2200 times smaller than that of FR and 1100 times smaller than that of Cor. At mid period, the fourth order error coefficient of 4B′ and 4ACB′ are too small to be extracted with confidence. Both can be well fitted with an eight order error term as shown in Figure 6.

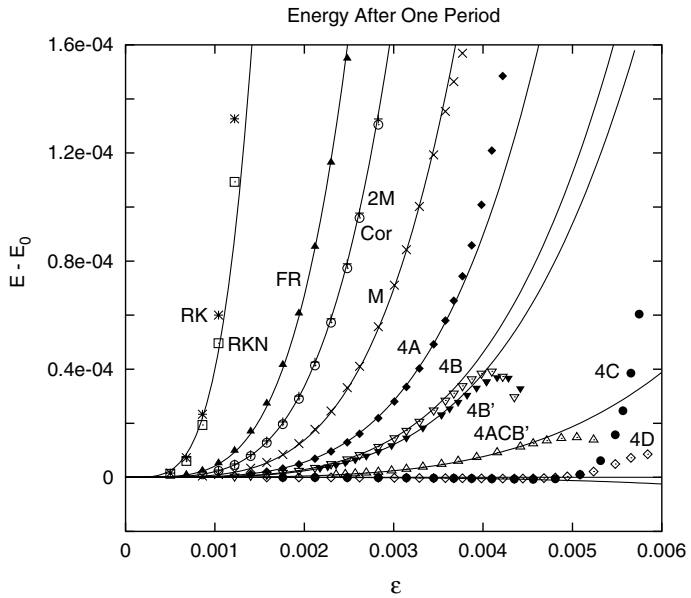

Figure 5. The convergence of the energy after one period. For the ease of comparison, the signs of 4B′, RK, and RKN have been changed from negative to positive. All can be very well fitted by solid lines of the form $c_i\epsilon^4$. Though not visible, the convergence curves for 4C and 4D definitely have a negative fourth order bend before turning positive. The plus sign corresponding to the kernel algorithm 2M, which is nearly indistinguishable from the complete corrector algorithm Cor denoted by hollow circles.

rector. This magic here is due to the special case of exactly one closed orbit. The correctors in Cor simply cancels out everywhere and kernel algorithm is fourth order by itself. For further discussion, see Chin (2004).

As in the Jacobi constant case, all forward symplectic algorithms perform better than non-forward ones. Interestingly, algorithm 4ACB′ is substantially better than 4A but not 4C. Algorithm 4C and 4D appear to be anomalous; their error is definitely fourth order, but with exceedingly small coefficients c_i. In Table I, we give the inverse of error coefficient $1/|c_i|$ normalized to that of RF. Thus the error coefficient of RK or RKN is ten times larger than that of RF, while the error coefficient of 4C is nearly 2200 times smaller than that of RF. To give a more quantitative comparison, one should consider the number of force and gradient evaluations. Although it is not unreasonable to assume that the gradient would require more effort than evaluating the force, when the force and gradient are evaluated at the same time, the additional effort to compute the gradient, as shown by (32), is minimal. A comparison with RF and M, which requires three and four force evaluations respectively, should give a reasonable gauge of FSI effectiveness.

Taking the fourth root of $1/|c_i|$ gives the effective step size relative to RF for attaining the same error. Thus in the case of 4C one can use steps size $(2200)^{1/4} \approx 7$ times as large as RF's and 4.6 times as large as M's.

Some of the observed anomalous behavior may be due to the fact that we have chosen an end point that is too "tame". After one period, the third body returns to a position that's far from both attractors, with no substantial error in the Jacobi constant. To test this hypothesis, we compute the energy again at mid period, when the third body has a close encounter and the error in the Jacobi constant is at a peak. This is shown in Figure 6. One immediately sees that in this case the kernel algorithm 2M converges only quadratically as it should and is distinct from the fourth order corrector algorithm Cor. Moreover, the convergence of both 4C and 4D is now clearly fourth order and bested by the optimized algorithm 4ACB′. However, fourth order fits to 4B′ and 4ACB′ are exceedingly unnatural, with very small coefficients. Both can be well fitted with a single *eight* order curve as shown. The inverse of the error coefficient is given in Table I. Despite some better than expected behaviors, it is clear from Table I that forward symplectic integrator as a

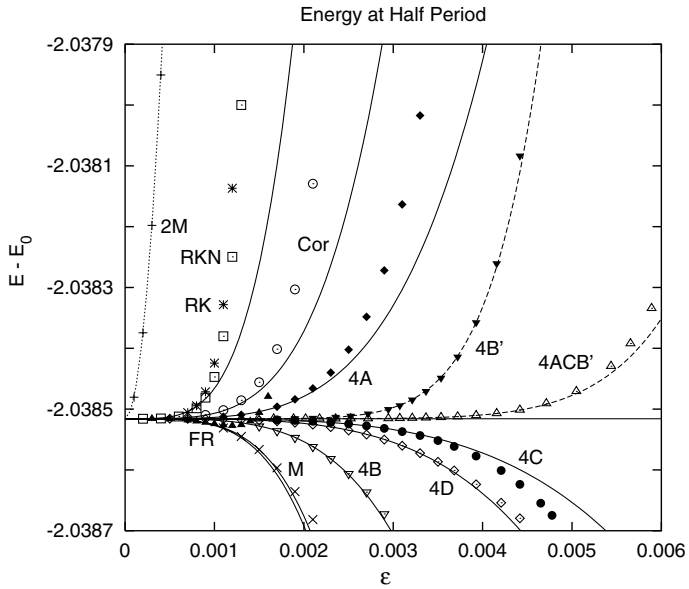

Figure 6. The convergence of the energy at mid period where the error in the Jacobi constant is at its peak. The solid lines are fourth order fits. The dotted line is a quadratic fit to the kernel algorithm 2M. The dashed lines are *eight* order fits to 4B′ and 4ACB′ whose fourth order coefficients are exceedingly small. The upright solid triangles are RF's results, which have an initial negative fourth order slope, very close to that of M, before turning positive.

class, can be orders of magnitude more efficient than non-forward integrators.

5. Conclusions

Forward symplectic algorithms are the only decomposition algorithms possible for solving time-irreversible problems (Auer et al. 2001; Forbert and Chin 2001a; Forbert and Chin 2001b; Ciftja and Chin 2003; Chin 2004). However, even for time-reversible problems, they have been shown to be more efficient than non-forward symplectic integrators (Chin 1997; Chin and Kidwell 2000; Chin and Chen 2001, 2002). In this work, we have further demonstrated their efficiency in solving classical dynamical problems with time-dependent forces. In the circular restricted three-body problem, this class of FSI can have errors orders of magnitude smaller than non-symplectic or non-forward algorithms. While FSI can be used for any classical calculations, such as molecular dynamics (Omelyan et al. 2002, 2003), they are particularly ideal for doing long time, high precision evolution of gravitational few-body problems. The force gradient is easily calculable and no more time consuming than evaluating the force. In contrast to other algorithms, FSI showed excellent, even better-than-expected, convergence behavior at close encounters. The existence of a parametrized family of algorithms allows one to optimize the algorithm for individual applications. This family of FSI should be applied to more realistic and more complex astrophysical problems. For planetary simulations without close-encounter, one can further improve this class of algorithm by incorporating Keplerian orbits (Wisdom and Holman 1991). This would produce a truly fourth order version [and not just a corrector version (Wisdom et al. 1996)] of the Wisdom–Holman algorithm.

While FSI have been largely overlooked in the development of classical symplectic integrators, they are precisely in accordance with McLachlan and Scovel's recommendation (McLachlan and Scovel 1996) that "derivative" symplectic algorithms should be developed. This work suggests that a systematic way of deriving higher order "derivative" algorithms is to devise factorization schemes that retain higher order commutators.

Process or corrector algorithms can achieve fourth order accuracy with only one evaluation of the force and its gradient. If 4A or $4B'$ are used as kernels, one should be able to achieve sixth order accuracy with only two evaluations of the force and one evaluation of the gradient. The use of fourth order FSI as kernel algorithms would generate a new family of sixth order process symplectic algorithms with minimal numbers of force evaluation. This work is currently in progress. If one does not insist on having purely

forward time steps, then many higher order algorithms can be generated on the basis of these fourth order FSI. See extensive constructions by Omelyan et al. (2002, 2003).

At this time, despite intense effort, no sixth order FSI has been found. Clearly research in FSI is still in its infancy and deserves further studies.

Acknowledgements

This work was supported, in part, by the National Science Foundation grants PHY-0100839 and DMS-0310580.

References

Albrecht, J.: 1955, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, AIAA, 1999, p. 603.

Auer, J., Krotscheck, E. and Chin, S. A.: 2001, 'A fourth-order real-space algorithm for solving local Schrodinger equations', *J. Chem. Phys.* **115**, 6841–6846.

Battin, R. H.: 1999, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, AIAA.

Blanes, S., Casas, F. and Ros, J.: 1999, 'Symplectic integrators with processing: A general study', *Siam J. Sci. Comput.* **21**, 711–727.

Chambers, J. and Murison, M. A.: 2000, 'Pseudo-high-order symplectic integrators', *Astron. J.* **119**, 425–433.

Campostrini, M. and Rossi, P.: 1990, 'A comparison of numerical algorithms for dynamical fermions', *Nucl. Phys.* B **329**, 753–764.

Candy, J. and Rozmus, W.: 1991, 'A symplectic integration algorithm for separable Hamiltonian functions', *J. Comp. Phys.* **92**, 230–256.

Channell, P. J. and Neri, F. R.: 1996, 'An introduction to symplectic integrators', Integration algorithms and classical mechanics (Toronto, ON, 1993), Fields Inst. Commun., 10, Amer. Math. Soc., Providence, RI, pp. 45–58.

Chin, S. A.: 1997, 'Symplectic integrators from composite operator factorizations', *Phys. Lett.* A **226**, 344–348.

Chin, S. A. and Kidwell, D. W.: 2000, 'Higher-order force gradient symplectic algorithms', *Phys. Rev.* E **62**, 8746–8752.

Chin, S. A. and Chen, C. R.: 2001, 'Fourth order gradient symplectic integrator methods for solving the time-dependent Schrodinger equation', *J. Chem. Phys.* **114**, 7338–7341.

Chin, S. A. and Chen, C. R.: 2002, 'Gradient symplectic algorithms for solving the Schrodinger equation with time-dependent potentials', *J. Chem. Phys.* **117**, 1409–1415.

Chin, S. A.: 2004, 'Quantum statistical calculations and symplectic corrector algorithms', *Phys. Rev.* E **69**, 046118.

Ciftja, O. and Chin, S. A.: 2003, 'Short-time-evolved wave functions for solving quantum many-body problems', *Phys. Rev.* B **68**, 134510.

Creutz, M. and Gocksch, A.: 1989, 'Higher-order hydrid Monte-Carlo algorithms', *Phys. Rev. Lett.* **63**, 9–12.

Forbert, H. A. and Chin, S. A.: 2001a, 'Fourth-order algorithms for solving the multivariable Langevin equation and the Kramers equation', *Phys. Rev.* E **63**, 016703.
Forbert, H. A. and Chin, S. A.: 2001b, 'Fourth-order diffusion Monte Carlo algorithms for solving quantum many-body problems', *Phys. Rev.* B **63**, 144518.
Forest, E. and Ruth, R. D.: 1990, '4th-order symplectic integration', *Physica D* **43**, 105–117.
Gladman, B., Duncan, M. and Candy, J.: 1991, 'Symplectic integrators for long-term integrations in celestial mechanics', *Celest. Mech. Dyn. Astron.* **52**, 221–240.
Goldman, D. and Kaper T. J.: 1996, 'Nth-order operator splitting schemes and nonreversible systems', *SIAM J. Numer. Anal.* **33**, 349–367.
Jang, S., Jang, S. and Voth, G. A.: 2001, 'Applications of higher order composite factorization schemes in imaginary time path integral simulations', *J. Chem. Phys.* **115**, 7832–7842.
Kinoshita, H., Yoshida, H. and Nakai, H.: 1991, 'Symplectic integrators and their application to dynamical astronomy', *Celest. Mech. Dyn. Astron.* **50**, 59–71.
Koseleff, P. V.: 1993, in *Applied algebra, algebraic algorithms and error-correcting codes* (San Juan, PR, 1993), Lecture Notes in Comput. Sci., Vol. 673, Springer, Berlin, p. 213
Koseleff, P. V.: 1996, In *Integration algorithms and classical mechanics*, Fields Inst. Commun., 10, Amer. Math. Soc., Providence, RI, p. 103
Laskar, J. and Robutel, P.: 2001, 'Higher order symplectic integrators for perturbed Hamiltonian systems', *Celest. Mech. Dyn. Astr.* **80**, 39–62.
Lopez-Marcos, M. A., Sanz-Serna, J. M., and Skeel R. D.: 1996, In: D. F. Griffiths and G. A. Watson, (eds), *Numerical Analysis 1995*, Longman, Harlow, UK, pp. 107–122.
Lopez-Marcos, M. A., Sanz-Serna, J. M. and Skeel R. D.: 1997, 'Explicit symplectic integrators using Hessian-vector products', *SIAM J. Sci. Comput.* **18**, 223–238.
McLachlan R. I. and Atela, P.: 1991, 'The accuracy of symplectic integrators' *Nonlinearity*, **5**, 541–562.
McLachlan, R. I.: 1995a, 'On the numerical integration of ordinary differential equations by symmetric composition methods', *SIAM J. Sci. Comput.* **16**, 151–168.
McLachlan, R. I.: 1995b, 'Composition methods in the presence of a small parameters', *BIT* **35**, 258–268.
McLachlan, R. I.: 1996, 'More on symplectic correctors', In: J. E. Marsden, G. W. Patrick, and W. F. Shadwick, (eds), *Integration Algorithms and Classical Mechanics*, American Mathematical Society, Providence, RI, pp. 141–149.
McLachlan, R. I. and Scovel, C.: 1996, 'Open problems in symplectic integration', *ibid*, pp. 151–180.
McLachlan, R. I., Reinout, G. and Quispel, W.: 2002, 'Splitting methods', *Acta Numerica,* **11**, 241–434.
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' *Phys. Rev.* E **66**, 026701.
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', *Comput. Phys. Commun.* **151**, 271–314.
Rowlands, G.: 1991, 'A numerical algorithm for Hamiltonian-systems', *J. Comput. Phys.* **97**, 235–239.
Ruth, R.: 1983, 'A canonical integration technique', *IEEE Trans. Nucl. Sci.* **30**, 2669–2671.
Scuro, S. and Chin, S. A.: 2004, *Forward Symplectic Integrators and the Long Time Phase Error in Periodic Motion*, arXiv, math-ph/0411086.

Sheng, Q.: 1989, 'Solving linear partial differential equations by exponential splitting', *IMA J. Numer. Anal.* **9**, 199–212.

Suzuki, M.: 1990, 'Fractal decomposition of exponential operators with applications to many-body theories and Monte-Carlo simulations', *Phys. Lett.* A **146**, 319–323.

Suzuki, M.: 1991, 'General theory of fractal path-integrals with applications to many-body theories and statistical physics', *J. Math. Phys.* **32**, 400–407.

Suzuki, M.: 1993, 'General decomposition-theory of ordered exponentials' *Proc. Jpn. Acad. Ser. B* **69**, 161–166.

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*, Springer, Berlin, pp. 1–6.

Wisdom, J. and Holman, M.: 1991, 'Symplectic maps for the N-body problem', *Astron. J.* **102**, 1528–1538.

Wisdom, J., Holman, M. and Touma, J.: 1996, 'Symplectic correctors', In: J. E. Marsden, G. W. Patrick, and W. F. Shadwick, (eds), *Integration Algorithms and Classical Mechanics*, American Mathematical Society, Providence, RI, pp. 217–244.

Yoshida, H.: 1990, 'Construction of higher order symplectic integrators', *Phys. Lett.* A **150**, 262–268.

Yoshida, H.: 1993, 'Recent progress in the theory and application of symplectic integrators', *Celest. Mech. Dyn. Astron.* **56**, 27–43.