Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 203 (2005) 250–273

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Almost symplectic Runge–Kutta schemes for Hamiltonian systems

Xiaobo Tan *

Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

Received 30 December 2003; received in revised form 19 May 2004; accepted 24 August 2004
Available online 21 September 2004

Abstract

Symplectic Runge–Kutta schemes for the integration of general Hamiltonian systems are implicit. In practice, one has to solve the implicit algebraic equations using some iterative approximation method, in which case the resulting integration scheme is no longer symplectic. In this paper, the preservation of the symplectic structure is analyzed under two popular approximation schemes, fixed-point iteration and Newton's method, respectively. Error bounds for the symplectic structure are established when N fixed-point iterations or N iterations of Newton's method are used. The implications of these results for the implementation of symplectic methods are discussed and then explored through numerical examples. Numerical comparisons with non-symplectic Runge–Kutta methods and pseudo-symplectic methods are also presented.

MSC: 65L05; 65L06; 65P10

Keywords: Geometric integrators; Hamiltonian structure; Symplectic Runge–Kutta methods; Pseudo-symplecticity; Fixed-point iteration; Newton's method; Convergence

1. Introduction

Geometric integration methods – numerical methods that preserve geometric properties of the flow of a differential equation – outperform off-the-shelf schemes (e.g., fourth-order explicit Runge–Kutta method) in predicting the long-term qualitative behaviors of the original system [6]. For systems evolving on differentiable manifolds (including the important setting of Lie groups), geometric integrators that preserve the

* Present address: Department of Electrical and Computer Engineering, Michigan State University, 2120 Engineering Building, East Lansing, MI 48824, USA. Tel.: +1 517 432 5671; fax: +1 517 353 1980.

E-mail address: xtan@egr.msu.edu.

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.08.012

manifolds are currently a subject of great interest to theorists and practitioners. See for instance [3]. Applications of such techniques are of interest in a variety of physical settings. See for instance [8] for results related to the integration of Landau–Lifshitz–Gilbert equation of micromagnetics.

An important class of geometric integrators are symplectic integration methods for Hamiltonian systems. See [14,9] and references therein. Symplectic integration algorithms have been used in many branches of physics. For instance, in the simulation of particle accelerators the conservation of the symplectic structure is so important that it motivated the development of the first symplectic schemes [12].

When the Hamiltonian has a separable structure, i.e., $H(p, q) = T(p) + V(q)$, explicit Runge–Kutta type algorithms exist which preserve the symplectic structure [5,17,4,10]. However, for general Hamiltonian systems, the symplectic Runge–Kutta schemes are implicit [13]. In practice, one has to solve the implicit algebraic equations for the intermediate stage values using some iterative approximation method such as fixed-point iteration or Newton's method.

In general, with an approximation based on a finite number of iterations, the resulting integration scheme is no longer symplectic. Error analysis on the structural conservation, like the analysis on the numerical accuracy, provides insight into a numerical method and helps in making judicious choices of integration schemes. An example of this is [2], where the error estimate for the Lie–Poisson structure was given for integration of Lie–Poisson systems using the mid-point rule. The first objective of this paper is to investigate the loss of symplectic structure due to the approximation in solving the implicit algebraic equations. The fixed-point iteration-based approximation and Newton's method-based approximation are analyzed, respectively. For either method, an error bound on the symplecticity of the numerical flow is established when N iterations are adopted for any $N \geqslant 1$. It turns out that, under suitable conditions, the convergence rate of the symplectic structure is closely related (but not equal) to the rate of convergence to the true solution of the implicit equations. Hence the methods become *almost symplectic* as N gets large.

The implications of the error bounds for implementing implicit, symplectic Runge–Kutta schemes are then studied in combination with a series of numerical examples. The question is how to strike the right balance between the computational cost and the structural preservation. Choices of the step size, the initial iteration value, and fixed point iteration versus Newton's method are discussed. Numerical comparisons are also conducted with a non-symplectic explicit Runge–Kutta method and with a pseudo-symplectic method proposed in [1]. Note that pseudo-symplectic integrators are explicit and designed to conserve the symplectic structure to a certain order.

The rest of the paper is organized as follows. In Section 2, the symplectic conditions for Runge–Kutta methods are first briefly reviewed to fix the notation, and then the fixed-point iteration-based approximation is analyzed. Analysis on Newton's method-based approximation is presented in Section 3. Comparisons among these approximation schemes and two other schemes are conducted in Section 4 through various numerical examples with a special focus on the nonlinear pendulum. Finally, some concluding remarks are provided in Section 5.

2. Fixed-point iteration-based approximation

2.1. Symplectic Runge–Kutta schemes

Consider a Hamiltonian system

$$\begin{cases} \dot{p}(t) = -\frac{\partial H(p,q)}{\partial q}, \\ \dot{q}(t) = \frac{\partial H(p,q)}{\partial p}, \end{cases} \tag{1}$$

with the Hamiltonian $H(p, q)$, where $(p, q) \in \mathbb{R}^d \times X$ for some integer $d \geqslant 1$, and X, the configuration space, is some d-dimensional manifold. In this paper, $X = \mathbb{R}^d$ is assumed for ease of discussion, but the extension of the results to the case of a general X is straightforward. Let $\ \triangleq \binom{p}{q}$. Then (1) can be rewritten as:

$$\dot{}(t) = f(\ (t)) \triangleq J\nabla H(\ (t)), \tag{2}$$

where

$$J = \begin{bmatrix} 0 & -I_d \\ I_d & 0 \end{bmatrix},$$

I_d denotes the d-dimensional identity matrix, and ∇_z stands for the gradient with respect to z.

An s-stage Runge–Kutta method to integrate (2) is as follows [7]:

$$\begin{cases} y_i = \ _0 + \tau\sum_{j=1}^{s} a_{ij} f(y_j), \quad i = 1, \ldots, s, \\ \ _1 = \ _0 + \tau\sum_{i=1}^{s} b_i f(y_i), \end{cases} \tag{3}$$

where τ is the step size, z_0 is the initial value at time t_0, z_1 is the numerical solution at time $t_0 + \tau$, a_{ij}, b_i are appropriate coefficients satisfying the order conditions of the Runge–Kutta method.

Let Ψ_τ be the mapping associated with the algorithm (3), i.e., $z_1 = \Psi_\tau(z_0)$. From [13], the transformation Ψ_τ preserves the symplecticity of the original system (2) if

$$b_i a_{ij} + b_j a_{\mathrm{ji}} - b_i b_j = 0, \quad i, j = 1, \ldots, s. \tag{4}$$

Thus if (4) is satisfied, we have:

$$\left(\frac{\partial \Psi_\tau}{\partial \ _0}\right)' J \left(\frac{\partial \Psi_\tau}{\partial \ _0}\right) - J = 0, \tag{5}$$

where "′" stands for the transpose. The condition (4) forces the symplectic Runge–Kutta method (3) to be implicit.

To put (3) in a more compact form, denote

$$\mathbf{y} \triangleq \begin{pmatrix} y_1 \\ \vdots \\ y_s \end{pmatrix}, \quad \mathbf{F}(\mathbf{y}) \triangleq \begin{pmatrix} f(y_1) \\ \vdots \\ f(y_s) \end{pmatrix},$$

$b \triangleq (b_1, \ldots, b_s)$, $A_0 \triangleq [a_{ij}]$, and $\mathbf{A} \triangleq A_0 \otimes I_{2d}$, where "⊗" denotes the Kronecker (tensor) product. Recall for two matrices $M = [m_{ij}]$ and $R = [r_{ij}]$, the Kronecker product

$$M \otimes R = \begin{bmatrix} m_{11}R & m_{12}R & \cdots \\ m_{21}R & m_{22}R & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}.$$

The algorithm (3) can now be written as

$$\begin{cases} \mathbf{y} = \mathbf{G}(\ _0, \mathbf{y}) \triangleq \mathbf{1} \otimes \ _0 + \tau\mathbf{A}\mathbf{F}(\mathbf{y}), \\ \ _1 = \ _0 + \tau b \otimes I_{2d}\mathbf{F}(\mathbf{y}), \end{cases} \tag{6}$$

where $\mathbf{1}$ is an s-dimensional column vector with 1 in every entry.

The results in this paper will make extensive use of norms (or induced norms) of vectors, matrices, and third-rank tensors. All norms will be denoted by $\|\cdot\|$, the specific meaning of which depends on the context. In particular, let $x = (x_1, \ldots, x_n)' \in \mathbb{R}^n$, $P \in \mathbb{R}^{p\times n}$ (a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^p$), and $Q \in \mathbb{R}^{n^2\times n}$ (a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^{n^2}$, i.e., a third-rank tensor). Then

$$\|x\| \triangleq \sqrt{\sum\nolimits_{i=1}^{n} x_i^2},$$
$$\|P\| \triangleq \sup_{x\in\mathbb{R}^n, x\neq 0} \frac{\|P\cdot x\|}{\|x\|} = \lambda_{\max}(P'P),$$
$$\|Q\| \triangleq \sup_{x\in\mathbb{R}^n, x\neq 0} \frac{\|Q\cdot x\|}{\|x\|},$$

where $\lambda_{\max}(P'P)$ denotes the largest eigenvalue of $P'P$, and "$\cdot$" denotes the action of a linear operator on a vector. When the operator is a matrix P, the action is the usual matrix multiplication and hereafter it will be just written as "Px".

Following the definitions, $\|Px\| \leqslant \|P\|\|x\|$, and $\|Q\cdot x\| \leqslant \|Q\|\|x\|$. The induced norms are *submultiplicative* (see [11, p. 410]): for $P_1 \in \mathbb{R}^{p\times k}$, $P_2 \in \mathbb{R}^{k\times n}$,

$$\|P_1P_2\| \leqslant \|P_1\|\|P_2\|.$$

It should be noted that although the Euclidean norm (and its induced norms) are used here, similar results with slight modifications could be obtained using other norms considering the equivalence of norms on finite-dimensional vector spaces.

2.2. Approximation based on fixed-point iteration

It is well-known that for a fixed z_0, when τ is sufficiently small, there is a unique solution $\mathbf{y}^*$ to the first equation in (6) and it can be obtained through fixed-point iteration [7]. The following proposition states a similar result; the key difference is that uniform convergence (with respect to z_0) is achieved. As we shall see, such uniform convergence is crucial for establishing the convergence of the symplectic structure.

For an open set $\Omega \subset \mathbb{R}^{2d}$, its ϵ-neighborhood, $\mathcal{N}(\Omega,\epsilon)$, is defined as

$$\mathcal{N}(\Omega,\epsilon) \triangleq \left\{ \quad \in \mathbb{R}^{2d} : \min_{\ _0\in\bar{\Omega}} \| \quad - \quad _0\| \leqslant \epsilon \right\},$$

where $\bar{\Omega}$ denotes the closure of Ω. Denote by $\mathcal{N}^s(\Omega,\epsilon)$ the product of s copies of $N(\Omega, \epsilon)$,

$$\mathcal{N}^s(\Omega,\epsilon) \triangleq \mathcal{N}(\Omega,\epsilon) \times \cdots \times N(\Omega,\epsilon).$$

Proposition 2.1. *Let $\Omega \subset \mathbb{R}^{2d}$ be a bounded, convex, open set. Let f be continuously differentiable. Then for any $\epsilon > 0$, there exists $\tau_0 > 0$ dependent on Ω and ϵ such that, $\forall\tau \leqslant \tau_0$, $\forall z_0 \in \Omega$,*

(1) $\mathbf{G}(z_0, \cdot)$ *maps $\mathcal{N}^s(\Omega,\epsilon)$ into itself;*
(2) *There is a unique solution* $\mathbf{y}^*$ *to the first equation in (6), and it can be approximated iteratively via*

$$\begin{cases} \mathbf{y}^{[n]} = \mathbf{G}(\ _0, \mathbf{y}^{[n-1]}), \\ \mathbf{y}^{[0]} = \mathbf{1} \otimes \quad _0 \end{cases} \tag{7}$$

and

(3) $\|\mathbf{y}^{[n]} - \mathbf{y}^*\| \leqslant \delta^n \|\mathbf{y}^{[0]} - \mathbf{y}^*\|$ *with* $0 < \delta < 1$, *where* $\delta = \tau C_1\|A_0\|$ *and* $C_1 \triangleq \max_{\ \in\mathcal{N}(\Omega,\epsilon)} \left\|\frac{\partial f}{\partial \ }(\ )\right\|$.

Proof. Denote $C_0 \triangleq \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon)}\|\mathbf{F}(\mathbf{y})\|$. Let $\tau_1 = \epsilon/(C_0\|A_0\|)$ (note that $\|A_0\| = \|\mathbf{A}\|$). Then $\forall\tau \leqslant \tau_1$, $\forall z_0\in\Omega$, $\mathbf{G}(z_0, \cdot)$ maps $\mathcal{N}^s(\Omega, \epsilon)$ into itself. Let $\tau_2 > 0$ be such that $\tau_2 C_1\|A_0\| < 1$. Since $\mathbf{G}(z_0, \cdot)$ is Lipschitz continuous with Lipschitz constant $\tau C_1\|A_0\|$ by the convexity assumption, it becomes a contraction mapping on $\mathcal{N}^s(\Omega, \epsilon)$ when $\tau \leqslant \tau_0 \triangleq \min\{\tau_1,\tau_2\}$. The rest of the claims then follows from the contraction mapping principle [16]. □

Remark 2.1. The convexity of Ω is assumed only for using the mean value theorem to get the estimate of Lipschitz constant. This assumption is not restrictive since one can resort to its convex hull if Ω is not convex.

An explicit but approximate algorithm to solve (6) is as follows: for some $N \geqslant 1$,

$$\begin{cases} \mathbf{y}^{[k]} = \mathbf{G}(\ {}_{0}, \mathbf{y}^{[k-1]}), \quad k = 1, \ldots, N, \\ \mathbf{y}^{[0]} = \mathbf{1} \otimes\ {}_{0}, \\ {}_{1}^{[N]} =\ {}_{0} + \tau b \otimes I_{2d}\mathbf{F}(\mathbf{y}^{[N]}). \end{cases} \tag{8}$$

From the implicit function theorem, when τ is sufficiently small, the solution $\mathbf{y}^*$ to the first equation in (6) is a function of z_0, written as $\mathbf{y}^*(z_0)$, and

$$\frac{\partial\mathbf{y}^*}{\partial\ {}_{0}}(\ {}_{0}) = \left[I_{2sd} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y}^*(\ {}_{0}))\right]^{-1}[\mathbf{1}\otimes I_{2d}]. \tag{9}$$

Similarly z_1 in (6), $\{\mathbf{y}^{[k]}\}_{k=0}^{N}$ and ${}_{1}^{[N]}$ in (8) (and smooth functions of them) are all continuously differentiable functions of z_0. In the sequel, when we write, e.g., $\partial\mathbf{y}^*/\partial z_0$ or $(\partial/\partial z_0)\mathbf{F}(\mathbf{y}^{[N]})$, *we think of* $\mathbf{y}^*$ *or* $\mathbf{F}(\mathbf{y}^{[N]})$ *as a function of* z_0 *although it is not explicitly written out.*

Denote by $\Psi_\tau^{[N]}$ the mapping associated with the algorithm (8), i.e., ${}_{1}^{[N]} = \Psi_\tau^{[N]}(\ {}_{0})$. The following lemma will be essential for studying how far $\Psi_\tau^{[N]}$ is away from being symplectic.

Lemma 2.1. *Let* $\Omega \subset \mathbb{R}^{2d}$ *be bounded, convex and open. For* $\epsilon > 0$, *pick* τ_0 *as in the proof of Proposition* 2.1. *Let f be twice continuously differentiable on* $\mathcal{N}(\Omega, \epsilon)$. *Then* $\forall\tau \leqslant \tau_0$, $\forall z_0 \in \Omega$,

$$\left\|\frac{\partial\mathbf{y}^{[N]}}{\partial\ {}_{0}} - \frac{\partial\mathbf{y}^*}{\partial\ {}_{0}}\right\| \leqslant \frac{D_0(C_1^2 + C_0C_2N)\delta^{N+1}}{C_1^2}, \tag{10}$$

$$\left\|\frac{\partial}{\partial\ {}_{0}}(\mathbf{F}(\mathbf{y}^{[N]}) - \mathbf{F}(\mathbf{y}^*))\right\| \leqslant \frac{D_0(C_1^2 + C_0C_2(1+N))\delta^{N+1}}{C_1}, \tag{11}$$

where $\delta \triangleq \tau C_1\|A_0\|$,

$$D_0 \triangleq \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon),\tau\leqslant\tau_0}\left\|\left[I_{2sd} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\right]^{-1}[\mathbf{1}\otimes I_{2d}]\right\|\left(= \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon),\tau\leqslant\tau_0}\sqrt{s}\left\|\left[I_{2\mathrm{sd}} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\right]^{-1}\right\|\right), \tag{12}$$

$$C_0 \triangleq \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon)}\|\mathbf{F}(\mathbf{y})\|,$$

$$C_1 \triangleq \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon)}\left\|\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\right\|\left(= \max_{\ \in\mathcal{N}(\Omega,\epsilon)}\left\|\frac{\partial f}{\partial\ }\right\|\right), \tag{13}$$

$$C_2 \triangleq \max_{\mathbf{y}_{i,j}\in\mathcal{N}^s(\Omega,\epsilon),1\leqslant i,j\leqslant 2sd}\|\mathbf{Q}(\{\mathbf{y}_{i,j}\})\|, \tag{14}$$

and $\mathbf{Q}(\{\mathbf{y}_{i,j}\})$ *is a third-rank tensor whose* (*i,j*)*th element is a vector given by* $\frac{\partial}{\partial \mathbf{y}}(\frac{\partial \mathbf{F}}{\partial \mathbf{y}})_{i,j}(\mathbf{y}_{i,j})$ *(here* $(\partial\mathbf{F}/\partial\mathbf{y})_{i,j}$ *denotes the* (*i,j*)*th component of* $\partial\mathbf{F}/\partial\mathbf{y}$).

Proof. See Appendix A. □

The main result of this section is:

Theorem 2.1. *Let* $\Omega \subset \mathbb{R}^{2d}$ *be bounded, convex and open. For* $\epsilon > 0$, *pick* τ_0 *as in the proof of Proposition* 2.1. *Let f be twice continuously differentiable on* $\mathcal{N}(\Omega,\epsilon)$. *Then* $\forall\tau \leqslant \tau_0$, $\forall z_0 \in \Omega$,

$$\left\|\left(\frac{\partial\Psi_\tau^{[N]}(\ _0)}{\partial\ _0}\right)' J\left(\frac{\partial\Psi_\tau^{[N]}(\ _0)}{\partial\ _0}\right) - J\right\| \leqslant \frac{2\|b\|D_0D_1(C_1^2 + C_0C_2(1+N))\delta^{N+2}}{\|A_0\|C_1^2} + \left(\frac{\|b\|D_0(C_1^2 + C_0C_2(1+N))\delta^{N+2}}{\|A_0\|C_1^2}\right)^2, \tag{15}$$

where

$$D_1 \triangleq \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon),\tau\leq\tau_0} \left\| I_{2d} + \tau b \otimes I_{2d}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\left[I_{2\mathrm{sd}} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\right]^{-1}[\mathbf{1}\otimes I_{2d}]\right\|, \tag{16}$$

and δ *and the other constants are as defined in Lemma* 2.1.

Proof. Let Ψ_τ be the mapping associated with (6). From (6) and (8),

$$\Lambda^{[N]}(\ _0) \triangleq \Psi_\tau^{[N]}(\ _0) - \Psi_\tau(\ _0) = \tau b \otimes I_{2d}(\mathbf{F}(\mathbf{y}^{[N]}) - \mathbf{F}(\mathbf{y}^*)).$$

Using Lemma 2.1, one derives

$$\left\|\frac{\partial\Lambda^{[N]}(\ _0)}{\partial\ _0}\right\| \leqslant \frac{\tau\|b\|D_0(C_1^2 + C_0C_2(1+N))\delta^{N+1}}{C_1}. \tag{17}$$

Next write

$$\begin{aligned}\left\|\left(\frac{\partial\Psi_\tau^{[N]}(\ _0)}{\partial\ _0}\right)' J\left(\frac{\partial\Psi_\tau^{[N]}(\ _0)}{\partial\ _0}\right) - J\right\| &= \left\|\left(\frac{\partial\Lambda^{[N]}(\ _0)}{\partial\ _0} + \frac{\partial\Psi_\tau(\ _0)}{\partial\ _0}\right)' J\left(\frac{\partial\Lambda^{[N]}(\ _0)}{\partial\ _0} + \frac{\partial\Psi_\tau(\ _0)}{\partial\ _0}\right) - J\right\| \\ &\leqslant \left\|\left(\frac{\partial\Lambda^{[N]}(\ _0)}{\partial\ _0}\right)' J\left(\frac{\partial\Lambda^{[N]}(\ _0)}{\partial\ _0}\right)\right\| + 2\left\|\left(\frac{\partial\Lambda^{[N]}(\ _0)}{\partial\ _0}\right)' J\left(\frac{\partial\Psi_\tau(\ _0)}{\partial\ _0}\right)\right\| \\ &\quad + \left\|\left(\frac{\partial\Psi_\tau(\ _0)}{\partial\ _0}\right)' J\left(\frac{\partial\Psi_\tau(\ _0)}{\partial\ _0}\right) - J\right\|,\end{aligned}$$

where the last term vanishes since Ψ_τ is symplectic. The claim now follows from (17), $\|J\| = 1$, and

$$\left\|\frac{\partial\Psi_\tau(\ _0)}{\partial\ _0}\right\| = \left\|I_{2d} + \tau b\otimes I_{2d}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y}^*)\frac{\partial\mathbf{y}^*}{\partial\ _0}\right\| \leqslant D_1. \qquad \square \tag{18}$$

Remark 2.2. Theorem 2.1 provides a structural error bound of $\Psi_\tau^{[N]}$ in terms of various constants specific to the problem of interest. Absorbing the constants and dropping the second term in the right-hand side of (15) (since the first term dominates), the error bound is simplified to $(c_1 + c_2N)\delta^{N+2}$ for $c_1, c_2 > 0$ and $0 < \delta < 1$. Note the connection and the difference between this bound and item 3 of Proposition 2.1. As N gets large, the structural error approaches zero and $\Psi_\tau^{[N]}$ becomes almost symplectic.

3. Newton's method-based approximation

Newton's method is an alternative to the fixed point iteration scheme for solving the implicit equation in (6). It reads

$$\mathbf{y}^{[n]} = \tilde{\mathbf{G}}(\ {}_0, \mathbf{y}^{[n-1]}) \triangleq \mathbf{y}^{[n-1]} - \left[I_{2\mathrm{sd}} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y}^{[n-1]})\right]^{-1}\left(\mathbf{y}^{[n-1]} - \mathbf{1}\otimes\ {}_0 - \tau\mathbf{A}\mathbf{F}(\mathbf{y}^{[n-1]})\right). \tag{19}$$

Typically convergence conditions for Newton's method include that the Jacobian is invertible at the solution point and that the initial condition is close enough to the solution [15]. Such conditions often cannot be verified directly. For the special case (6), however, Proposition 3.1 shows that when taking the natural candidate for $\mathbf{y}^{[0]}$, the convergence is guaranteed if $\tau < \tau_0$, where τ_0 can be determined explicitly.

Proposition 3.1. *Let $\Omega \subset \mathbb{R}^{2d}$ be a bounded, convex, open set. Let f be three times continuously differentiable. Then for any $\epsilon > 0$, there exists $\tau_0 > 0$ dependent on Ω and ϵ such that, $\forall\tau \leqslant \tau_0$, $\forall z_0 \in \Omega$,*

(1) $\tilde{\mathbf{G}}(\ {}_0, \cdot)$ *maps $\mathcal{N}^s(\Omega, \epsilon)$ into itself;*
(2) *There is a unique solution $\mathbf{y}^*$ to the first equation in (6), and it can be approximated iteratively via*

$$\begin{cases} \mathbf{y}^{[n]} = \tilde{\mathbf{G}}(\ {}_0, \mathbf{y}^{[n-1]}), \\ \mathbf{y}^{[0]} = \mathbf{1}\otimes\ {}_0, \end{cases} \tag{20}$$

and

(3) $\|\mathbf{y}^{[n]} - \mathbf{y}^*\| \leqslant K^{2^n-1}\|\mathbf{y}^{[0]} - \mathbf{y}^*\|^{2^n}$, *where $K > 0$ and $K\|\mathbf{y}^* - y^{[0]}\| < 1$.*

Proof. Through algebraic manipulations, $\tilde{\mathbf{G}}(\ {}_0, \mathbf{y})$ can be rewritten as

$$\tilde{\mathbf{G}}(\ {}_0, \mathbf{y}) = \mathbf{1}\otimes\ {}_0 + \tau\left[I_{2\mathrm{sd}} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\right]^{-1}\mathbf{A}\left[\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})(\mathbf{1}\otimes\ {}_0 - \mathbf{y}) + \mathbf{F}(\mathbf{y})\right]. \tag{21}$$

Pick $\tau_1 > 0$ such that $I_{2sd} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})$ is invertible $\forall\tau \leqslant \tau_1$, $\forall\mathbf{y} \in \mathcal{N}^s(\Omega, \epsilon)$. Let

$$E_0 \triangleq \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon), \tau\leqslant\tau_1}\left\|\left[I_{2sd} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\right]^{-1}\right\|, \tag{22}$$

$$E_1 \triangleq \max_{\mathbf{y}\in\mathcal{N}^s(\Omega,\epsilon),\ {}_0\in\Omega}\left\|\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})(\mathbf{1}\otimes\ {}_0 - \mathbf{y}) + \mathbf{F}(\mathbf{y})\right\|, \tag{23}$$

and let $\tau_2 > 0$ be such that $\tau_2 E_0 E_1 \|A_0\| < \epsilon$. Then it can be verified that if $\tau \leqslant \min\{\tau_1, \tau_2\}$, $\tilde{\mathbf{G}}(\ {}_0, \cdot)$ maps $\mathcal{N}^s(\Omega, \epsilon)$ into itself.

The next goal is to establish that $\tilde{\mathbf{G}}(\ {}_0, \cdot)$ is a contraction mapping. This can be done by evaluating $\frac{\partial\tilde{\mathbf{G}}}{\partial\mathbf{y}}$. To properly handle the third-rank tensor $\partial^2\mathbf{F}/\partial\mathbf{y}^2$ involved, for $\eta \in \mathbb{R}^{2sd}$, one calculates using (19)

$$\frac{\partial\tilde{\mathbf{G}}}{\partial\mathbf{y}}(\ {}_0, \mathbf{y})\eta = -\tau\mathbf{H}(\mathbf{y})\mathbf{A}\left(\frac{\partial^2\mathbf{F}}{\partial\mathbf{y}^2}(\mathbf{y})\cdot\eta\right)\mathbf{H}(\mathbf{y})[\mathbf{y} - \mathbf{1}\otimes\ {}_0 - \tau A\mathbf{F}(\mathbf{y})], \tag{24}$$

where

$$\mathbf{H}(\mathbf{y}) \triangleq \left[I_{2sd} - \tau\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})\right]^{-1}. \tag{25}$$

Eq. (24) implies

$$\left\|\frac{\partial\tilde{\mathbf{G}}}{\partial\mathbf{y}}(\ _0,\mathbf{y})\right\| \leqslant \tau\|\mathbf{H}(\mathbf{y})\|^2\|A_0\|\left\|\frac{\partial^2\mathbf{F}}{\partial\mathbf{y}^2}(\mathbf{y})\right\|\|\mathbf{y}-\mathbf{1}\otimes\ _0-\tau\mathbf{AF}(\mathbf{y})\|. \tag{26}$$

Denote

$$E_2 \triangleq \max_{\mathbf{y}\in\mathscr{N}^s(\Omega,\epsilon)}\left\|\frac{\partial^2\mathbf{F}}{\partial\mathbf{y}^2}(\mathbf{y})\right\|, \tag{27}$$

$$E_3 \triangleq \max_{\mathbf{y}\in\mathscr{N}^s(\Omega,\epsilon),\ _0\in\Omega,\tau\leqslant\tau_1}\|\mathbf{y}-\mathbf{1}\otimes\ _0-\tau\mathbf{AF}(\mathbf{y})\|, \tag{28}$$

and pick $\tau_3 > 0$ such that $\tau_3E_0^2E_2E_3\|A_0\| < 1$. Then when $\tau \leqslant \min\{\tau_1,\tau_2,\tau_3\}$, $\tilde{\mathbf{G}}(\ _0,\cdot)$ is a contraction mapping and hence (20) converges to a (unique) fixed point, which is the solution to the first equation in (6).

Since $\frac{\partial\tilde{\mathbf{G}}}{\partial\mathbf{y}}(\ _0,\mathbf{y}^*)=0$, the convergence rate of (20) is quadratic, as is standard for Newton's method [15]:

$$\|\mathbf{y}^{[n]}-\mathbf{y}^*\| \leqslant K\|\mathbf{y}^{[n-1]}-\mathbf{y}^*\|^2 \leqslant K^{2^n-1}\|\mathbf{y}^{[0]}-\mathbf{y}^*\|^{2^n}, \tag{29}$$

where

$$K \triangleq \max_{\mathbf{y}\in\mathscr{N}^s(\Omega,\epsilon),\ _0\in\Omega,\tau\leqslant\tau_1}\left\|\frac{\partial^2\tilde{\mathbf{G}}}{\partial\mathbf{y}^2}(\ _0,\mathbf{y})\right\|. \tag{30}$$

It's easy to see that $\frac{\partial^2\tilde{\mathbf{G}}}{\partial\mathbf{y}^2}(\ _0,\mathbf{y})$ contains a factor of τ. On the other hand, $\|\mathbf{y}^{[0]}-\mathbf{y}^*\| \leqslant \tau C_0\|A_0\|$, where C_0 is as defined in Lemma 2.1. Therefore there exists $\tau_4 > 0$ such that when $\tau \leqslant \tau_4$, $K\|\mathbf{y}^*-y^{[0]}\| < 1$. Finally τ_0 in the statement of the proposition can be chosen to be $\tau_0 = \min\{\tau_1,\tau_2,\tau_3,\tau_4\}$. □

Analogous to (8), an approximation scheme for solving (6) can be constructed based on Newton's method: for some $N \geqslant 1$,

$$\begin{cases}\mathbf{y}^{[k]} = \tilde{\mathbf{G}}(\ _0,\mathbf{y}^{[k-1]}), \quad k=1,\ldots,N,\\ \mathbf{y}^{[0]} = \mathbf{1}\otimes\ _0,\\ {}^{[N]}_{1} = \ _0+\tau b\otimes I_{2d}\mathbf{F}(\mathbf{y}^{[N]}).\end{cases} \tag{31}$$

Denote by $\tilde{\Psi}_\tau^{[N]}$ the mapping associated with the algorithm (31). The following two lemmas will be used in the proof of Theorem 3.1.

Lemma 3.1. *Let $\Omega \subset \mathbb{R}^{2d}$ be bounded, convex and open. For $\epsilon > 0$, pick τ_0 as in the proof of Proposition 3.1. Let f be three times continuously differentiable on $\mathscr{N}(\Omega,\epsilon)$. Define $\mathbf{H}(\cdot)$ as in (25), and $\mathbf{J}(\mathbf{y}) \triangleq \mathbf{H}(\mathbf{y})\mathbf{A}\frac{\partial\mathbf{F}}{\partial\mathbf{y}}(\mathbf{y})$. Then $\forall\tau\leqslant\tau_0$, $\forall z_0\in\Omega$,*

$$\left\|\frac{\partial\mathbf{y}^{[N]}}{\partial\ _0}\right\| \leqslant C_y \triangleq \sqrt{s}\left(1+\frac{E_0}{1-\gamma_0}\right), \tag{32}$$

$$\left\|\frac{\partial}{\partial\ _0}\mathbf{H}(\mathbf{y}^{[N]})\right\| \leqslant C_H \triangleq \frac{\gamma_0C_y}{E_3}, \tag{33}$$

$$\|\frac{\partial}{\partial\ _0}\mathbf{J}(\mathbf{y}^{[N]})\| \leqslant C_J \triangleq \frac{\|A_0\|(C_1\gamma_0+E_0E_2E_3)C_y}{E_3}, \tag{34}$$

where $\gamma_0 \triangleq \tau_0E_0^2E_2E_3\|A_0\|$; C_1 is as defined in (13); E_1, E_2 are as defined in (23), (27); and E_0 and E_3 are as defined in (22), (28) with τ_1 replaced by τ_0.

Proof. See Appendix B. □

Lemma 3.2. *Let $\Omega \subset \mathbb{R}^{2d}$ be bounded, convex and open. For $\epsilon > 0$, pick τ_0 as in the proof of Proposition* 3.1. *Let f be three times continuously differentiable on $\mathcal{N}(\Omega, \epsilon)$. Then $\forall\tau \leqslant \tau_0$, $\forall z_0 \in \Omega$,*

$$\left\| \frac{\partial \mathbf{y}^{[N]}}{\partial \;_0} - \frac{\partial \mathbf{y}^*}{\partial \;_0} \right\| \leqslant D_y \delta^{2^{N-1}}, \tag{35}$$

$$\left\| \frac{\partial}{\partial \;_0} \mathbf{F}(\mathbf{y}^{[N]}) - \frac{\partial}{\partial \;_0} \mathbf{F}(\mathbf{y}^*) \right\| \leqslant C_1 D_y \delta^{2^{N-1}} + \frac{C_2 D_0}{K} \delta^{2^N}, \tag{36}$$

where $\delta \triangleq \tau C_0 \|A_0\| K < 1$, $D_y \triangleq \frac{\tau_0}{K}(C_J + C_1 C_H \|A_0\| + \frac{1}{\sqrt{s}} C_2 D_0^2 \|A_0\|)$, C_J and C_H are as defined in Lemma 3.1, *and C_1, C_2, D_0 and K are as defined in (13), (14), (12) and (30), respectively.*

Proof. See Appendix C. □

Following the arguments as in the proof of Theorem 2.1 and using Lemma 3.2, we can show:

Theorem 3.1. *Let $\Omega \subset \mathbb{R}^{2d}$ be bounded, convex and open. For $\epsilon > 0$, pick τ_0 as in the proof of Proposition* 3.1. *Let f be three times continuously differentiable on $\mathcal{N}(\Omega, \epsilon)$. Let $\tilde{\Psi}_\tau^{[N]}$ be the mapping associated with* (31). *Then $\forall\tau \leqslant \tau_0$, $\forall z_0 \in \Omega$,*

$$\begin{aligned} \left\| \left(\frac{\partial \tilde{\Psi}_\tau^{[N]}(\;_0)}{\partial \;_0} \right)' J \left(\frac{\partial \tilde{\Psi}_\tau^{[N]}(\;_0)}{\partial \;_0} \right) - J \right\| &\leqslant 2\tau D_1 \|b\| \left(C_1 D_y \delta^{2^{N-1}} + \frac{C_2 D_0}{K} \delta^{2^N} \right) \\ &\quad + \left(\tau \|b\| \left(C_1 D_y \delta^{2^{N-1}} + \frac{C_2 D_0}{K} \delta^{2^N} \right) \right)^2, \end{aligned} \tag{37}$$

where D_1 is as defined in (16), *and δ and the other constants are as defined in Lemma* 3.2.

4. Numerical examples and discussion

The performances of approximation schemes (8) and (31) on symplectic structure conservation have been characterized in Theorems 2.1 and 3.1, respectively. Under suitable conditions and with proper choices for the step size and the initial iteration value $\mathbf{y}^{[0]}$, both schemes *uniformly* (with respect to z_0) converge, and the convergence rate of symplectic structure for either scheme is closely connected to the corresponding rate for the solution convergence (i.e., $\|\mathbf{y}^{[N]} - \mathbf{y}^*\|$). In this section, the implications of these results for implementing implicit, symplectic Runge–Kutta schemes are explored through a variety of numerical examples.

Important factors in choosing a Runge–Kutta scheme for Hamiltonian systems include the numerical accuracy, the structural preservation performance (symplecticity) and the computational cost. Since the issue of numerical accuracy is not the focus of this paper, the discussion will be centered around the interplay between the symplecticity and the computational complexity. For illustrative purposes, the methods listed

Table 1
Runge–Kutta methods used in numerical examples

Notation	Method	Order	Pseudo-symp. order	s
MidPoint	Mid-point rule	2	Symplectic	1
Gauss4	Gauss method [6]	4	Symplectic	2
PS63	Pseudo-symp. method [1]	3	6	5
RK4	Classical Runge–Kutta	4	4	4

Table 2
Test problems used in the numerical study

Problem	Hamiltonian $H(p,q)$	Step size τ	Initial condition
Nonlinear pendulum	$\frac{p^2}{2} - \cos(q)$	See the text	See the text
Linear pendulum	$\frac{1}{2}(p^2 + q^2)$	0.5	$(2,2)'$
Kepler problem	$\frac{1}{2}(p_1^2 + p_2^2) - \frac{1}{\sqrt{q_1^2+q_2^2}}$	$\frac{\pi}{64}$	$(0,2,0.4,0)'$
Bead on a wire	$\frac{p^2}{2(1+U'(q)^2)} + U(q)$ with $U(q) = 0.1(q(q-2))^2 + 0.008q^3$	$\frac{1}{6}$	$(0.49,0)'$
Galactic dynamics	$\frac{1}{2}(p_1^2 + p_2^2 + p_3^2) + \frac{1}{4}(p_1q_2 - p_2q_1) + \ln(1 + \frac{q_1^2}{a^2} + \frac{q_2^2}{b^2} + \frac{q_3^2}{c^2})$, with $a = \frac{5}{4}, b = 1, c = \frac{3}{4}$	0.2	$(0,1.689,0.2,2.5,0,0)'$

in Table 1 will be compared in the numerical problems. For a definition of pseudo-symplecticity order, we refer to [1]. The mid-point rule and the Gauss method are implicit, and both fixed-point iteration and Newton's method will be used to solve the implicit equations. Table 2 lists the test problems. Some of these problems were also used in [1]. The computation was done in Matlab on a Dell laptop Inspiron 4150.

4.1. The nonlinear pendulum problem

An essential property of a symplectic map is the preservation of the sum of oriented, projected areas onto the coordinate planes (p_i,q_i), $i = 1, \ldots, d$. For the nonlinear pendulum problem, $(p,q) \in \mathbb{R} \times \mathbb{R}$ and the projected area is just the phase space area. The ellipse shown in Fig. 1(a), with semi-major axis $r_{\mathrm{maj}} = 1.8$ and semi-minor axis $r_{\mathrm{min}} = 1.2$, encloses the (continuous) set S_0 of initial conditions for this problem. The area occupied by S_0 is $\mathscr{A}_0 = \pi r_{\mathrm{maj}} r_{\mathrm{min}}$. Given an integration scheme, the set S_0 evolves, say, into another S_1 at time t with area $\mathscr{A}_1$. The (normalized) area change is then defined as

$$\delta^* \triangleq \frac{|\mathscr{A}_1 - \mathscr{A}_0|}{\mathscr{A}_0}.$$

Since in general it is impossible to evaluate $\mathscr{A}_1$ exactly, an approximation scheme is introduced by first discretizing the ellipse into $\bar{n}$ points (see Fig. 1(a) for illustration with $\bar{n} = 8$). Then $\mathscr{A}_1$ is approximately equal

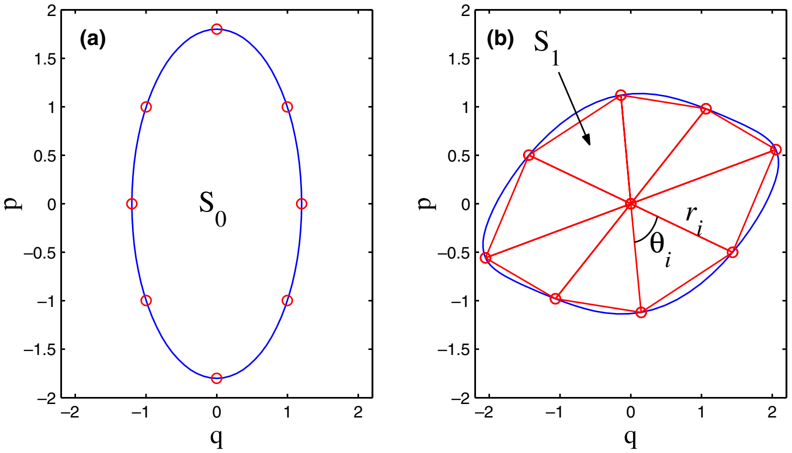


Fig. 1. (a) Initial conditions for the nonlinear pendulum problem; (b) Approximating the area of S_1 with a finite number of triangles.

to the sum $\hat{\mathscr{A}}_1$ of areas of the triangles formed by the $\bar{n}$ solution points at time t and the origin (Fig. 1 (b)). Define the *approximate* (normalized) area change as

$$\hat{\delta} \triangleq \frac{|\hat{\mathscr{A}}_1 - \mathscr{A}_0|}{\mathscr{A}_0}.$$

It is of interest to estimate the error $|\hat{\delta} - \delta^*|$. When $\bar{n}$ is large, the ith triangle in Fig. 1(b) is almost isosceles with side r_i and vertex angle $\theta_i \approx \theta_{\bar{n}} \triangleq \frac{2\pi}{\bar{n}}$. The area of the ith triangle is thus approximated by $\frac{1}{2} r_i^2 \sin(\theta_i)$. The corresponding portion of S_1 is approximately a circular sector with radius r_i and vertex angle θ_i, the area of which is $\frac{1}{2} r_i^2 \theta_i$. Therefore,

$$\frac{|\hat{\mathscr{A}}_1 - \mathscr{A}_1|}{\mathscr{A}_1} \approx \frac{\sum_{i=1}^{\bar{n}} \frac{r_i^2}{2} |\sin(\theta_i) - \theta_i|}{\sum_{i=1}^{\bar{n}} \frac{r_i^2}{2} \theta_i} \approx \frac{\sum_{i=1}^{\bar{n}} \frac{r_i^2}{2} |\sin(\theta_{\bar{n}}) - \theta_{\bar{n}}|}{\sum_{i=1}^{\bar{n}} \frac{r_i^2}{2} \theta_{\bar{n}}} = \frac{|\sin(\theta_{\bar{n}}) - \theta_{\bar{n}}|}{\theta_{\bar{n}}}.$$

Let $\epsilon_{\bar{n}} \triangleq \frac{|\sin(\theta_{\bar{n}}) - \theta_{\bar{n}}|}{\theta_{\bar{n}}}$. Writing $\hat{\delta} = \frac{|\mathscr{A}_1 - \mathscr{A}_0 + \hat{\mathscr{A}}_1 - \mathscr{A}_1|}{\mathscr{A}_0}$ and considering $\frac{\mathscr{A}_1}{\mathscr{A}_0} \approx 1$, one can see that a bound estimate for $|\ \hat{\delta} - \delta^*\ |$ is $\epsilon_{\bar{n}}$.

From the above analysis, $\epsilon_{\bar{n}}$ can be thought of as the accuracy of the area approximation scheme. If $\hat{\delta} < \epsilon_{\bar{n}}$, one can only infer that δ^* is close to $\epsilon_{\bar{n}}$ but cannot link the specific value of $\hat{\delta}$ to δ^*. For this purpose, in plotting the numerical results these data points will be set to $\hat{\delta} = \epsilon_{\bar{n}}$ with a distinct symbol. In the computational results to be reported next, $\bar{n} = 10^5$, and $\epsilon_{\bar{n}} = 6.58 \times 10^{-10}$. This choice of $\bar{n}$ has been found to offer a good tradeoff between the area approximation accuracy and the computational cost.

Results in Sections 2 and 3 can provide guidance in selecting step sizes to guarantee the uniform convergence. For instance, consider MidPoint for the nonlinear pendulum problem. It can be shown that for any $\tau < 2$, the fixed-point iteration converges. For Newton's method, it is more involved to compute the *maximum* step size that ensures the uniform convergence; on the other hand, it is relatively easy to establish convergence for $\tau \leqslant 0.2$. Thus both schemes would converge if one chooses $\tau = 0.2$. However, for $\tau = 0.2$, the computed area change $\hat{\delta}$ after one step falls below $\epsilon_{\bar{n}}$ when the iteration number $N = 2$ for Newton's method, preventing one from getting a meaningful $\hat{\delta}$ versus N curve. Therefore, in the following simulation $\tau = 1.6$, 0.8, and 0.2 are used, where the uniform convergence is numerically verified. Note that a step size as big as 1.6 might be too large if one is concerned about the numerical accuracy of solutions. However, here the numerical accuracy is not a concern and the emphasis is on investigating how the area change $\hat{\delta}$ varies with the iteration number N in solving the implicit equations.

To get a qualitative feel about the area-preservation performances of the four methods listed in Table 1, numerical solutions after one step are obtained with these methods and are compared with the exact solution, see Fig. 2. Here $\tau = 1.6$, and the implicit equations in MidPoint and Gauss4 were solved using Newton's method up to machine accuracy. As one can see, the (exact) final configuration is distorted from the initial elliptical curve. By the symplecticity of the exact flow, the area enclosed by the exact solutions at $t = 1.6$ is equal to that enclosed by the ellipse at $t = 0$. Among the numerical solutions, Gauss4 has the best performance in terms of accuracy and area-preservation since it completely overlaps the exact solution. The solution of MidPoint is noticeably different from that of the exact one because it is of the second order. The area-preserving performance of MidPoint cannot be easily told from the figure (theoretically it should be as good as that of Gauss4). Under PS63 it can be seen that the area has shrunk a little bit, while RK4 delivers the worst performance in area preservation.

One goal of this paper is to provide insight into the choice of fixed-point iteration versus Newton's method. From Theorems 2.1 and 3.1, Newton's method enjoys much faster structural convergence than the fixed-point iteration in terms of the number of iterations. This is verified in Figs. 3 and 4. Fig. 3 shows the decrease of area change with the number of fixed-point iterations, where the underlying algorithm used

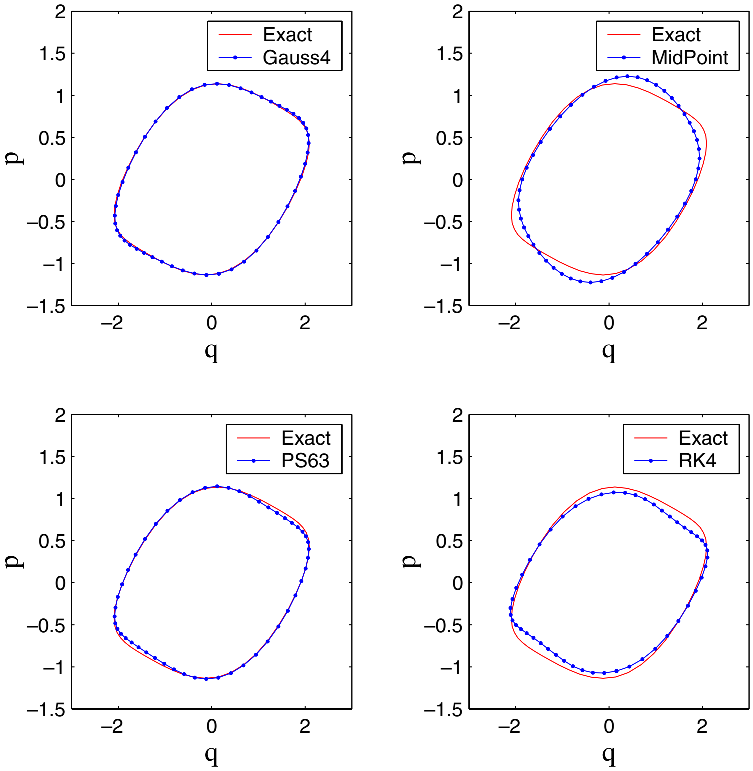


Fig. 2. Comparison of numerical solutions with the exact solution after one step ($\tau = 1.6$) for the nonlinear pendulum problem.

was MidPoint. In the figure, the bound from Theorem 2.1 is also plotted. Note the similar trend in both curves, in particular, their consistent convergence rates. For Newton's method, the area change reaches $\epsilon_{\bar{n}}$ within 4 iterations (Fig. 4).

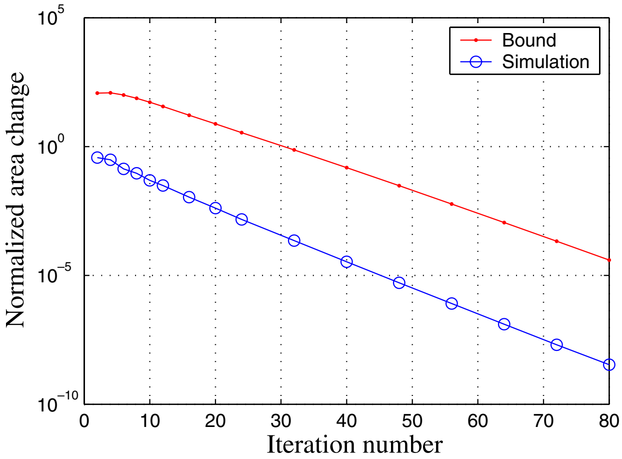


Fig. 3. Decrease of the area change $\hat{\delta}$ (one step) vs the number N of iterations for the nonlinear pendulum problem. MidPoint used with fixed-point iteration ($\tau = 1.6$).

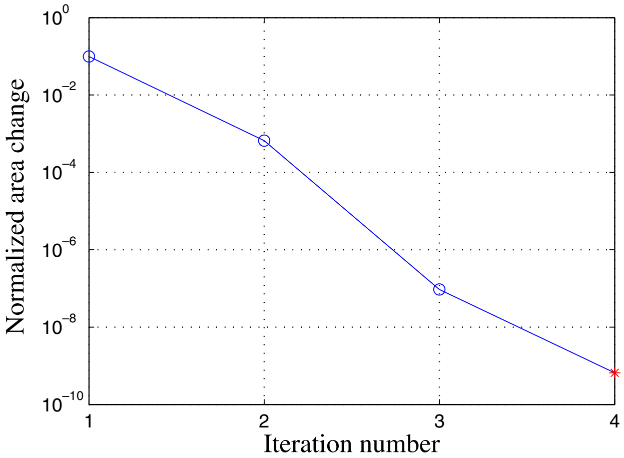


Fig. 4. Decrease of the area change $\hat{\delta}$ (one step) vs the number N of iterations for the nonlinear pendulum problem. MidPoint used with Newton's method ($\tau = 1.6$). For $N = 4$, $\hat{\delta} < \epsilon_{\bar{n}}$ as represented by the "*" symbol.

Despite the faster convergence, Newton's method takes longer time in each iteration than the fixed-point iteration. This brings up the issue whether the aforementioned advantage is still an advantage when the actual computational time is considered. In terms of N, the computational times of the two methods can be approximately expressed as $T_0^a + NT_1^a$, $T_0^b + NT_1^b$, respectively. Here T_0^a and T_1^a represent the computational overhead and the computational cost per iteration for the fixed-point scheme, respectively, and T_0^b and T_1^b represent the counterparts for Newton's method. The actual computation times taken by the two methods are plotted in Fig. 5, both displaying a linearly increasing trend. As N gets large, the ratio of their computation costs approaches a constant $\frac{T_1^a}{T_1^b}$. Considering their convergence rates, one can conclude that Newton's method is more time-efficient when very low structural error is needed.

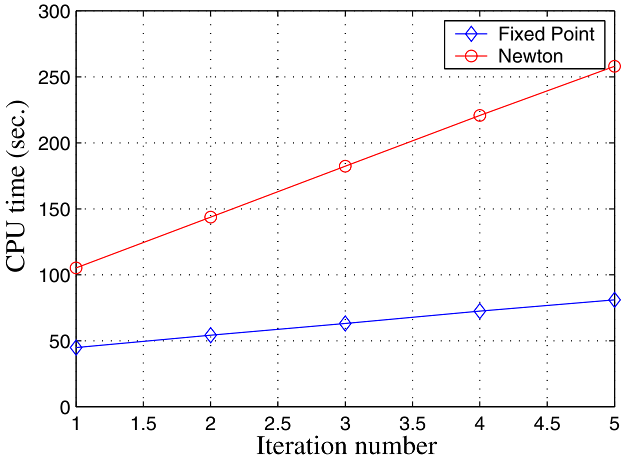


Fig. 5. Comparison of the computation time (for one step) vs the number N of iterations for fixed-point iteration and Newton's method. The nonlinear pendulum problem computed and MidPoint used with $\tau = 1.6$.

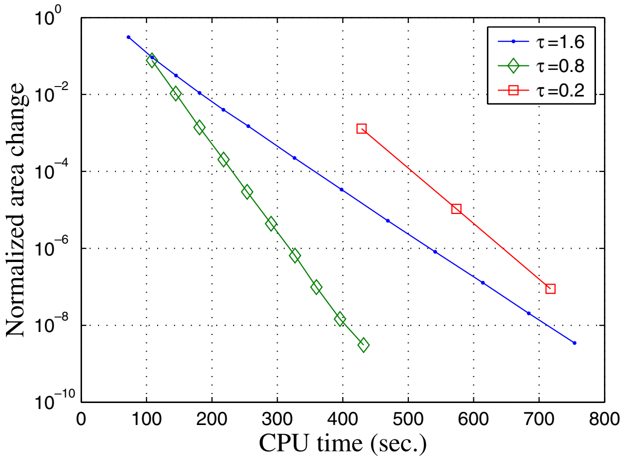


Fig. 6. Work-precision diagrams for the nonlinear pendulum problem under the fixed-point iteration scheme with different step sizes. Final time $t = 1.6$ fixed. Underlying algorithm: MidPoint.

Two other step sizes $\tau = 0.8$, $\tau = 0.2$ are used to integrate the nonlinear pendulum equation while the final time $t = 1.6$ is kept fixed. Therefore the total numbers of steps for these step sizes are 2 and 8, respectively. Fig. 6 shows the work-precision diagrams of the fixed-point iteration scheme for the three different step sizes. It can be seen that for the same amount of CPU time, with $\tau = 0.8$, the area change is smaller than that with $\tau = 1.6$ or with $\tau = 0.2$. It can be explained as follows: when τ is relatively big, the convergence rate is slow; while when τ is relatively small, it requires many steps which, to keep the total CPU time the same, leads to a small number N of iterations at each step. Therefore to maximize the computational efficiency (defined as the level of structural preservation per CPU time unit), one needs to seek a moderate step size. Fig. 7 shows the work-precision diagrams of Newton's method-based approximation under different step sizes. For this particular problem, even with $\tau = 1.6$, at most 4 iterations would bring the area

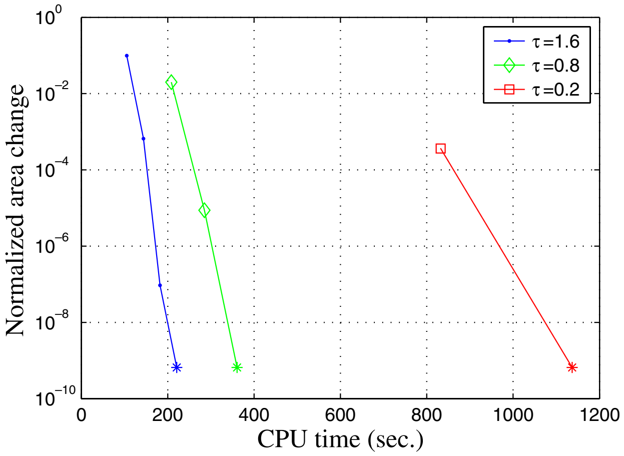


Fig. 7. Work-precision diagrams for the nonlinear pendulum problem under Newton's method-based scheme with different step sizes. Final time $t = 1.6$ fixed. Underlying algorithm: MidPoint. Computed $\hat{\delta} < \epsilon_{\bar{n}}$ for the data points represented by "*".

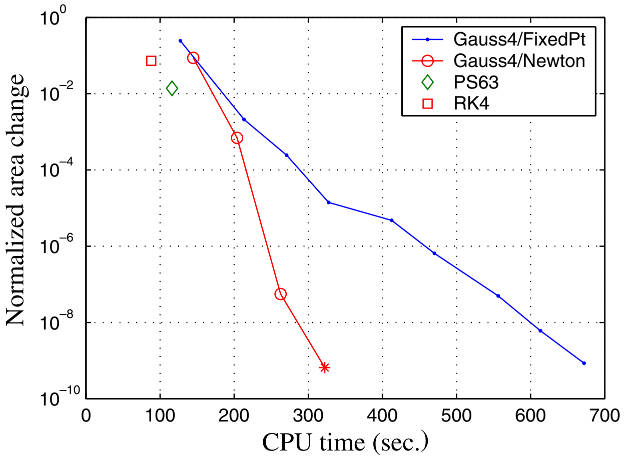


Fig. 8. Comparison of work-precision diagrams under different schemes ($\tau = 1.6$) for the nonlinear pendulum problem. Final time $t = 1.6$. Computed $\hat{\delta} < \epsilon_{\bar{n}}$ for the data point represented by "*".

change below the accuracy level $\epsilon_{\bar{n}}$, and there is not much to gain by using smaller τ in the sense of computational efficiency defined above.

Fig. 8 through Fig. 10 compare the work-precision diagrams of Gauss4/FixedPt (solving Gauss4 with fixed-point iteration), Gauss4/Newton (solving Gauss4 with Newton's method), PS63 and RK4 for different step sizes. PS63 always beats RK4 at a slight cost of computational time. For same amount of CPU time, PS63 also leads to smaller area change than Gauss4/FixedPt and Gauss4/Newton. However, while the structural error under Gauss4/FixedPt or Gauss4/Newton approaches zero with increasing CPU time, the error of PS63 can be large when τ is relatively big (Figs. 8 and 9). Finally, it can be seen that corresponding to relatively large error, Gauss4/FixedPt needs less CPU time than Gauss4/Newton; but for very small error, Gauss4/Newton requires less CPU time than Gauss4/FixedPt. Hence, again, Newton's method will

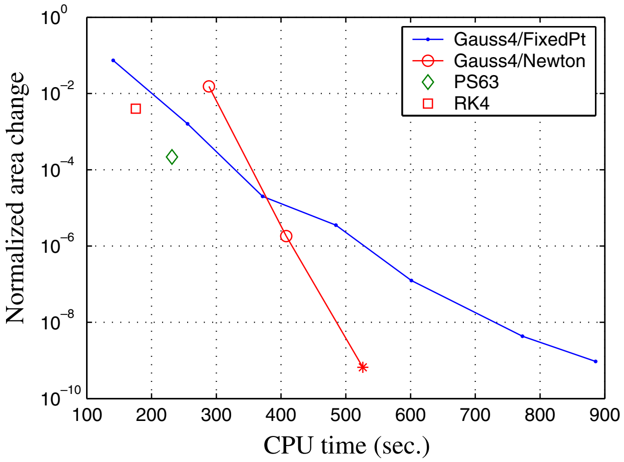


Fig. 9. Comparison of work-precision diagrams under different schemes ($\tau = 0.8$) for the nonlinear pendulum problem. Final time $t = 1.6$. Computed $\hat{\delta} < \epsilon_{\bar{n}}$ for the data point represented by "*".

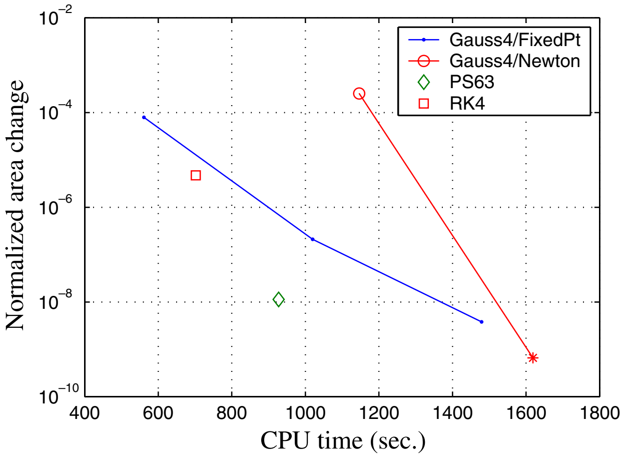


Fig. 10. Comparison of work-precision diagrams under different schemes ($\tau = 0.2$) for the nonlinear pendulum problem. Final time $t = 1.6$. Computed $\hat{\delta} < \epsilon_{\bar{n}}$ for the data point represented by "*".

be more competitive than the fixed-point iteration in long-time simulation, where the area change per step needs to be very small.

From (29) and the proof of Lemma 3.2, a better choice of $\mathbf{y}^{[0]}$ (i.e., smaller $\|\mathbf{y}^{[0]} - \mathbf{y}*\|$ with $\mathbf{y}^{[0]}$ smoothly dependent on z_0) leads to faster convergence of the symplectic structure. A hybrid approximation scheme is motivated by this observation: first use $\mathbf{1} \otimes z_0$ as the initial guess and run the fixed-point iteration N_1 times, then use $\mathbf{y}^{[N_1]}$ as the initial value and run Newton's method for N_2 iterations. The idea is to use relatively cheap computation of the fixed-point algorithm to get a better initial estimate for Newton's method. Fig. 11 shows the work-precision comparison of this hybrid scheme (with $N_1 = 1$) with the plain Newton's method, where both cases of $\tau = 1.6$ and $\tau = 0.8$ are displayed. From the figure it can be seen that the hybrid scheme offers faster convergence rate with a slight increase of computational cost.

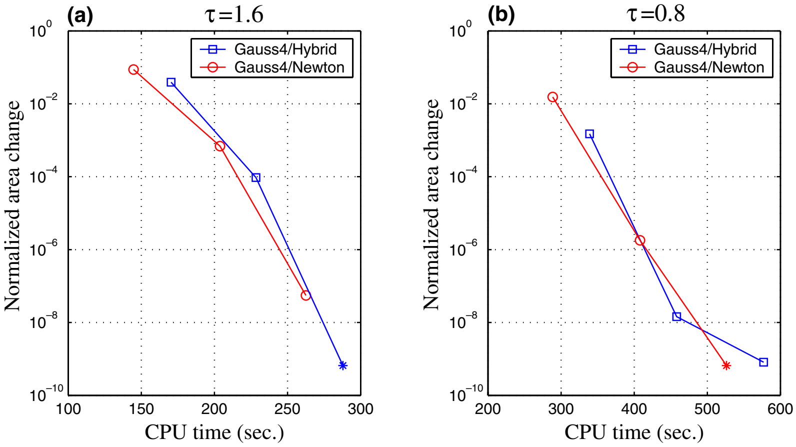


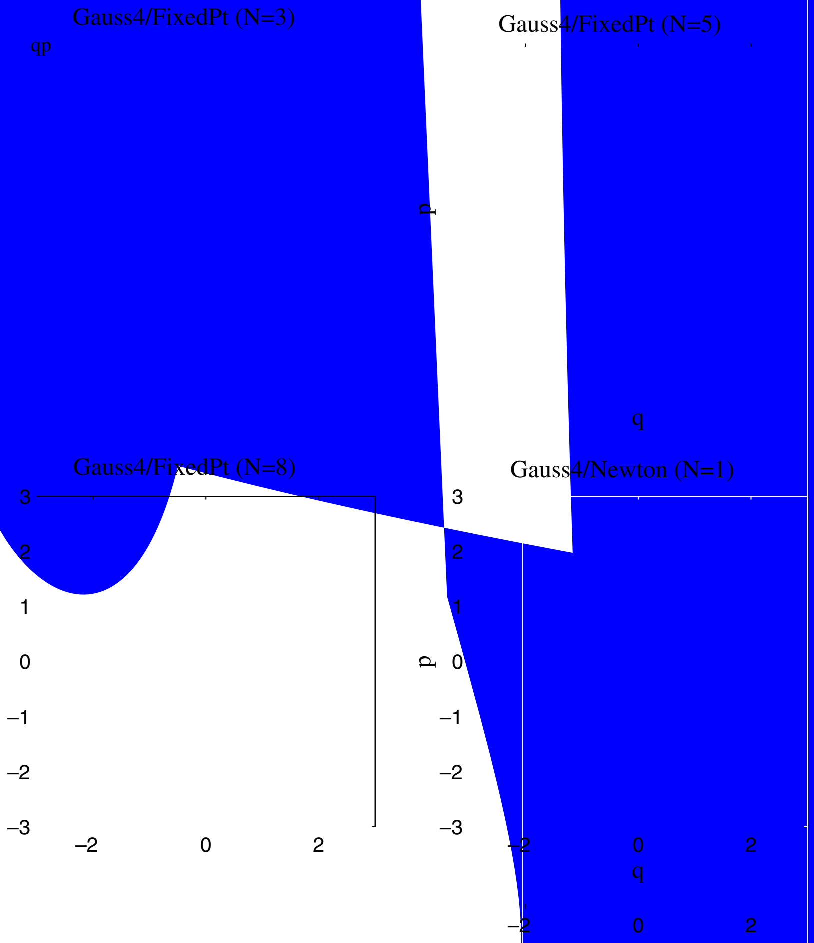
Fig. 11. Comparison of work-precision diagrams under the hybrid scheme and Newton's method for the nonlinear pendulum problem. Gauss4/Hybrid: run fixed point iteration once and then run Newton's method. (a) $\tau = 1.6$; (b) $\tau = 0.8$. Final time $t = 1.6$ for both (a) and (b). Computed $\hat{\delta} < \epsilon_{\bar{n}}$ for the data points represented by "*".

4.2. C

Th the performan vatio or the linear pe the f < 1.58 (Newto one p size used is pro

lum in the ph G $\times 10^4$ (the dat th energy decays T is increased t n the other han

and q_2 com orbit is also



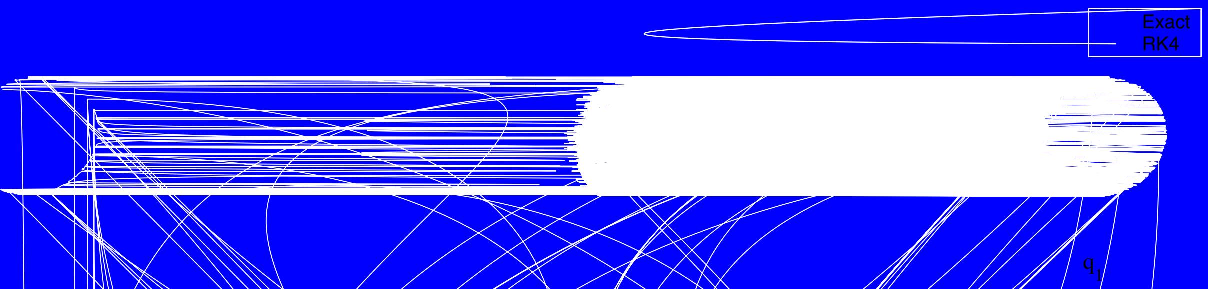


Fig. 13. Exact and numerical solutions of the Kepler problem. The underlying algorithm for FixedPt and Newton: Gauss4.

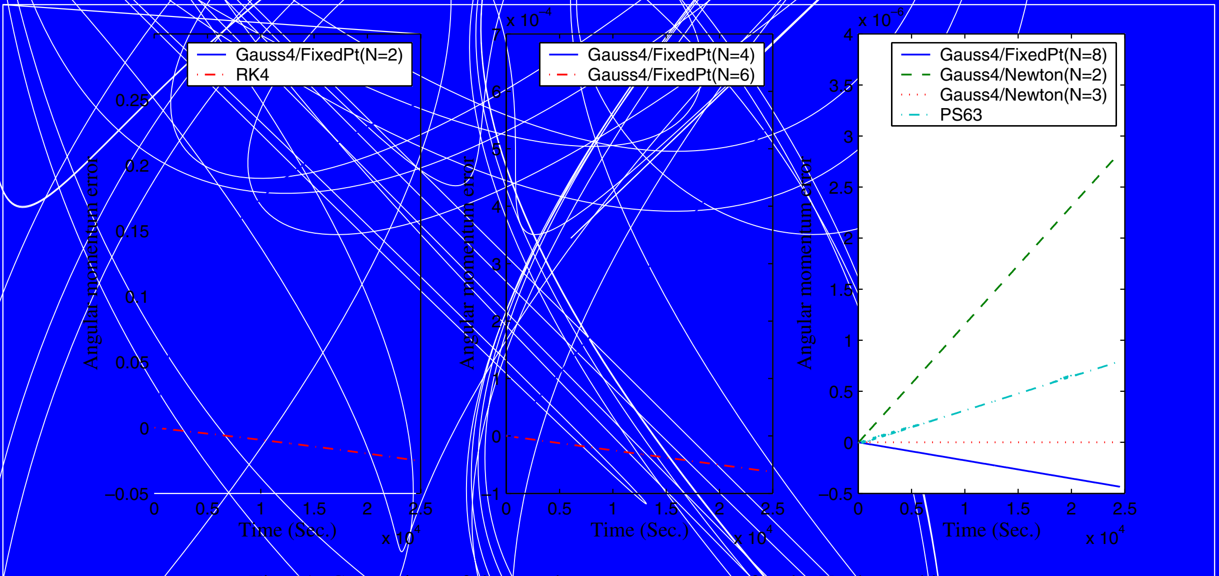


Fig. 14. Comparison of the angular momentum error for the Kepler problem.

Table 3
CPU time used in solving the Kepler problem

Method	Gauss4/FixedPt					Gauss4/Newton			PS63	RK4
	N	2	4	6	8	*N*	2	3		
Time (s)		777.7	1218.9	1651.0	2081.1		1395.5	1768.1	969.5	726.0

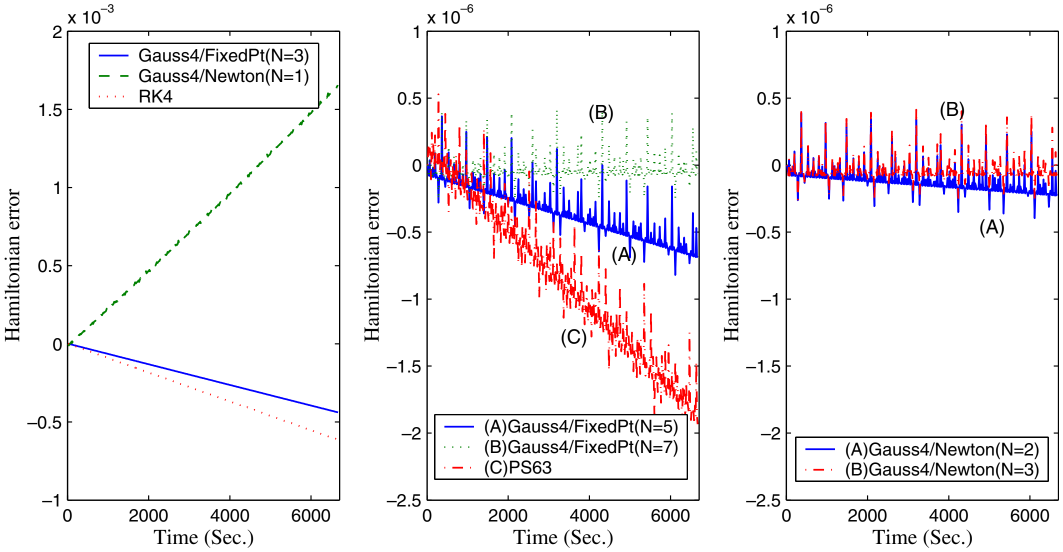


Fig. 15. Comparison of the Hamiltonian error for the bead-on-a-wire problem.

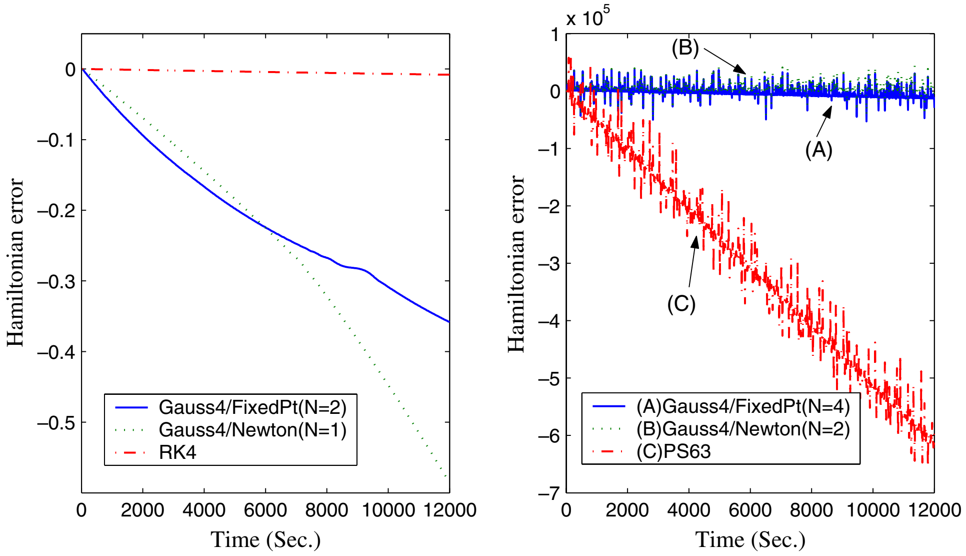


Fig. 16. Comparison of the Hamiltonian error for the galactic dynamics problem.

Table 4
CPU time used in solving the bead-on-a-wire problem

Method	Gauss4/FixedPt				Gauss4/Newton				PS63	RK4
	N	3	5	7	N	1	2	3		
Time (s)		160.4	197.0	240.0		158.3	191.0	220.4	146.4	129.2

Table 5
CPU time used in solving the galactic dynamics problem

Method	Gauss4/FixedPt			Gauss4/Newton			PS63	RK4
	N	2	4	N	1	2		
Time (s)		283.9	360.0		311.9	376.4	318.2	264.2

the ellipse. The angular momentum is a conserved quantity for the Kepler problem. Fig. 14 shows the angular momentum error under different schemes. Listed in Table 3 is the CPU time used in the computation.

Figs. 15 and 16 show the evolution of error in the Hamiltonian for the bead-on-a-wire problem and the galactic dynamics problem, where the total numbers of steps are 4×10^4 and 6×10^4, respectively. Tables 4 and 5 list the CPU time used by different algorithms.

5. Conclusions

Symplectic Runge–Kutta schemes for the integration of general Hamiltonian systems are implicit. When approximation methods are used to solve the implicit equations, the resulting integration schemes do not fully preserve the symplectic structure of the original systems. It is thus of interest to understand the structural error incurred by the approximation schemes. In this paper approximations based on two common iterative methods for solving implicit equations, fixed-point iteration and Newton's method, were analyzed and the corresponding error bounds established. Under proper conditions, these schemes become almost symplectic as the iteration number N gets large. Although the results show that the structural convergence of either scheme is closely related to its numerical convergence, the former (essentially $\| \frac{\partial \mathbf{y}^{[N]}}{\partial_0} - \frac{\partial \mathbf{y}^*}{\partial_0} \|$) does not follow merely from the latter ($\|\mathbf{y}^{[N]} - \mathbf{y}^*\|$); instead it is a consequence of the uniform convergence of the iterative schemes with respect to the initial condition z_0, the particular choices of the initial iteration values, and the smoothness of the mappings $\mathbf{G}(\cdot,\cdot)$ and $\tilde{\mathbf{G}}(\cdot,\cdot)$.

The theoretical results can be used in selecting an appropriate approximation scheme when integrating a specific problem. The emphasis here is the trade-off between the computational cost and the structural preservation performance although the numerical accuracy (the order of a method) also plays an important role in implementation. The faster convergence rate of Newton's method-based scheme makes it more favorable than the fixed-point iteration-based scheme, especially when very small structural error is required. This was verified in the numerical tests.

The effect of the step size on the computational efficiency was studied in the numerical experiments. We also note that the arguments in the proofs of Propositions 2.1 and 3.1 may be used to find the step size τ_0 (below which the scheme is uniformly convergent) for the specific problem of interest. For stiff problems, τ_0 will be very small for the fixed-point algorithm and Newton's method is generally more efficient. After observing that a better initial guess would speed up the convergence rate of Newton's method, a hybrid scheme (running one or several fixed-point iterations to obtain initial values for Newton's method) was proposed and explored. Simulation suggested that the hybrid scheme has a potential to out-perform the plain Newton's method.

The almost symplectic schemes were also compared against a pseudo-symplectic method and a non-symplectic method. It is of no surprise that the non-symplectic method delivers the poorest performance in area-conservation and energy-conservation. For methods of comparable orders of accuracy, the pseudo-symplectic one delivers slightly better structural preserving performance than an approximation-based symplectic scheme *if* the latter spends the same amount of CPU time. However, with increased CPU time (which is still comparable to the CPU time used by the pseudo-symplectic one) the approximation scheme has the potential to reach very low structural error and becomes almost symplectic. On the other hand, as admitted in [1], the design of a pseudo-symplectic method (in particular, of order p and of pseudo-symplecticity order $2p$ [1]) beyond order (3,6) is very complicated. This will hinder the use of pseudo-symplectic methods in very long time simulation of Hamiltonian systems.

Acknowledgments

The work in this paper was supported in part by the Army Research Office under the ODDR&E MURI97 Program Grant No. DAAG55-97-1-0114 to the Center for Dynamics and Control of Smart Structures (through Harvard University) and under the ODDR&E MURI01 Program Grant No. DAAD19-01-1-0465 to the Center for Networked Communicating Control Systems (through Boston University), and by the Lockheed Martin Chair Endowment Funds. The author would like to thank Professor P.S. Krishnaprasad for numerous discussions and valuable suggestions on this work, and thank the anonymous referees for many constructive comments that significantly improved the paper.

Appendix A. Proof of Lemma 2.1

Proof. From (6) and (8),

$$\mathbf{y}^{[N]} - \mathbf{y}^* = \tau \mathbf{A}\big(\mathbf{F}(\mathbf{y}^{[N-1]}) - \mathbf{F}(\mathbf{y}^*)\big). \tag{A.1}$$

Taking derivative on both sides of (A.1) with respect to z_0 and re-arranging terms, one gets

$$\frac{\partial \mathbf{y}^{[N]}}{\partial \ _0} - \frac{\partial \mathbf{y}^*}{\partial \ _0} = \tau \mathbf{A}\left[\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]})\left(\frac{\partial \mathbf{y}^{[N-1]}}{\partial \ _0} - \frac{\partial \mathbf{y}^*}{\partial \ _0}\right) + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\right)\frac{\partial \mathbf{y}^*}{\partial \ _0}\right]. \tag{A.2}$$

Eq. (A.2) implies

$$\begin{aligned}\left\|\frac{\partial \mathbf{y}^{[N]}}{\partial \ _0} - \frac{\partial \mathbf{y}^*}{\partial \ _0}\right\| &\leqslant \tau\|A_0\|\left\|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]})\right\|\left\|\frac{\partial \mathbf{y}^{[N-1]}}{\partial \ _0} - \frac{\partial \mathbf{y}^*}{\partial \ _0}\right\| + \tau\|A_0\|\left\|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\right\|\left\|\frac{\partial \mathbf{y}^*}{\partial \ _0}\right\| \\ &\leqslant \tau C_1\|A_0\|\left\|\frac{\partial \mathbf{y}^{[N-1]}}{\partial \ _0} - \frac{\partial \mathbf{y}^*}{\partial \ _0}\right\| + \tau D_0\|A_0\|\left\|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\right\|. \end{aligned} \tag{A.3}$$

By the mean value theorem, the (*i,j*)th component of $\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)$ can be expressed as

$$\left(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\right)_{i,j} = \frac{\partial}{\partial \mathbf{y}}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}\right)_{i,j}(\mathbf{y}_{i,j}) \cdot (\mathbf{y}^{[N-1]} - \mathbf{y}^*) \quad \text{for some } \mathbf{y}_{i,j} \in \mathcal{N}^s(\Omega, \epsilon),$$

which leads to

$$\left\|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\right\| \leqslant C_2\|\mathbf{y}^{[N-1]} - \mathbf{y}^*\| \leqslant C_2\delta^{N-1}\|\mathbf{y}^{[0]} - \mathbf{y}^*\| \quad \text{(from Proposition 2.1)}$$
$$\leqslant \tau C_0C_2\|A_0\|\delta^{N-1} \text{ (since } \mathbf{y}^* - \mathbf{y}^{[0]} = \tau\mathbf{AF}(\mathbf{y}^*)). \tag{A.4}$$

Plugging (A.4) into (A.3), one has

$$\left\|\frac{\partial \mathbf{y}^{[N]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| \leqslant \delta\left\|\frac{\partial \mathbf{y}^{[N-1]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| + \frac{C_0C_2D_0\delta^{N+1}}{C_1^2}. \tag{A.5}$$

Performing recursions on (A.5), one gets

$$\begin{aligned}
\left\|\frac{\partial \mathbf{y}^{[N]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| &\leqslant \delta\left[\delta\left\|\frac{\partial \mathbf{y}^{[N-2]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| + \frac{C_0C_2D_0\delta^{N}}{C_1^2}\right] + \frac{C_0C_2D_0\delta^{N+1}}{C_1^2} \\
&= \delta^2\left\|\frac{\partial \mathbf{y}^{[N-2]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| + \frac{2C_0C_2D_0\delta^{N+1}}{C_1^2} \\
&\leqslant \delta^2\left[\delta\left\|\frac{\partial \mathbf{y}^{[N-3]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| + \frac{C_0C_2D_0\delta^{N-1}}{C_1^2}\right] + \frac{2C_0C_2D_0\delta^{N+1}}{C_1^2} \\
&= \delta^3\left|\frac{\partial \mathbf{y}^{[N-3]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| + \frac{3C_0C_2D_0\delta^{N+1}}{C_1^2} \\
&\ldots \\
&\leqslant \delta^N\left\|\frac{\partial \mathbf{y}^{[0]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| + \frac{C_0C_2D_0N\delta^{N+1}}{C_1^2}.
\end{aligned}$$

Eq. (10) is then proved by noting

$$\left\|\frac{\partial \mathbf{y}^{[0]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| = \left\|\tau\mathbf{A}\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right\| \leqslant \tau C_1D_0\|A_0\|.$$

To show (11), write

$$\frac{\partial}{\partial\,{}_0}\left(\mathbf{F}(\mathbf{y}^{[N]}) - \mathbf{F}(\mathbf{y}^*)\right) = \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N]})\left(\frac{\partial \mathbf{y}^{[N]}}{\partial\,{}_0} - \frac{\partial \mathbf{y}^*}{\partial\,{}_0}\right) + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\right)\frac{\partial \mathbf{y}^*}{\partial\,{}_0}, \tag{A.6}$$

and then use (10) and (A.4). □

Appendix B. Proof of Lemma 3.1

Proof. Differentiating both sides of (19) with respect to z_0 leads to

$$\frac{\partial \mathbf{y}^{[N]}}{\partial\,{}_0} = \frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{y}}(\,{}_0, \mathbf{y}^{[N-1]})\frac{\partial \mathbf{y}^{[N-1]}}{\partial\,{}_0} + \frac{\partial \tilde{\mathbf{G}}}{\partial\,{}_0}(\,{}_0, \mathbf{y}^{[N-1]}). \tag{B.1}$$

From $\|\frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{y}}(\,{}_0, \mathbf{y}^{[N-1]})\| \leqslant \tau E_0^2E_2E_3\|A_0\|$ (recall (26)) and $\|\frac{\partial \tilde{\mathbf{G}}}{\partial\,{}_0}(\,{}_0, \mathbf{y}^{[N-1]})\| = \|\mathbf{H}(\mathbf{y}^{[N-1]})\mathbf{1} \otimes I_{2d}\| \leqslant \sqrt{s}E_0$, one gets

$$\left\|\frac{\partial \mathbf{y}^{[N]}}{\partial\ {}_0}\right\| \leqslant \tau E_0^2 E_2 E_3 \|A_0\| \left\|\frac{\partial \mathbf{y}^{[N-1]}}{\partial\ {}_0}\right\| + \sqrt{s}E_0 = \gamma \left\|\frac{\partial \mathbf{y}^{[N-1]}}{\partial\ {}_0}\right\| + \sqrt{s}E_0, \tag{B.2}$$

where $\gamma \triangleq \tau E_0^2 E_2 E_3 \|A_0\|$. Performing recursions on (B.2) yields

$$\begin{aligned}
\left\|\frac{\partial \mathbf{y}^{[N]}}{\partial\ {}_0}\| \leqslant \gamma[\gamma\right\|\frac{\partial \mathbf{y}^{[N-2]}}{\partial\ {}_0}\| + \sqrt{s}E_0] + \sqrt{s}E_0 &= \gamma^2 \left\|\frac{\partial \mathbf{y}^{[N-2]}}{\partial\ {}_0}\right\| + (1+\gamma)\sqrt{s}E_0 \\
&\leqslant \gamma^2\left[\gamma\left\|\frac{\partial \mathbf{y}^{[N-3]}}{\partial\ {}_0}\right\| + \sqrt{s}E_0\right] + (1+\gamma)\sqrt{s}E_0 = \gamma^3\left\|\frac{\partial \mathbf{y}^{[N-3]}}{\partial\ {}_0}\right\| + (1+\gamma+\gamma^2)\sqrt{s}E_0 \\
&\ldots \\
&\leqslant \gamma^N \left\|\frac{\partial \mathbf{y}^{[0]}}{\partial\ {}_0}\right\| + \left(\sum_{n=0}^{N-1}\gamma^n\right)\sqrt{s}E_0 = \gamma^N \left\|\frac{\partial \mathbf{y}^{[0]}}{\partial\ {}_0}\right\| + \frac{\sqrt{s}E_0(1-\gamma^N)}{1-\gamma} \\
&= \sqrt{s}\left(\gamma^N + \frac{E_0(1-\gamma^N)}{1-\gamma}\right) \quad \left(\text{since} \left\|\frac{\partial \mathbf{y}^{[0]}}{\partial\ {}_0}\right\| = \sqrt{s}\right).
\end{aligned}$$

Eq. (32) then follows from $0 < \gamma \leqslant \gamma_0 < 1$.

Eq. (33) can be shown by writing

$$\frac{\partial}{\partial\ {}_0}\mathbf{H}(\mathbf{y}^{[N]}) = \frac{\partial \mathbf{H}}{\partial \mathbf{y}}(\mathbf{y}^{[N]})\frac{\partial \mathbf{y}^{[N]}}{\partial\ {}_0} = \tau \mathbf{H}(\mathbf{y}^{[N]})\mathbf{A}\frac{\partial^2 \mathbf{F}}{\partial \mathbf{y}^2}(\mathbf{y}^{[N]})\mathbf{H}(\mathbf{y}^{[N]})\frac{\partial \mathbf{y}^{[N]}}{\partial\ {}_0}$$

and then using (32).

Finally to show (34), note that

$$\frac{\partial}{\partial\ {}_0}\mathbf{J}(\mathbf{y}^{[N]}) = \frac{\partial}{\partial\ {}_0}\mathbf{H}(\mathbf{y}^{[N]})\mathbf{A}\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N]}) + \mathbf{H}(\mathbf{y}^{[N]})\mathbf{A}\frac{\partial^2 \mathbf{F}}{\partial \mathbf{y}^2}(\mathbf{y}^{[N]})\frac{\partial \mathbf{y}^{[N]}}{\partial\ {}_0},$$

and then use (32) and (33). □

Appendix C. Proof of Lemma 3.2

Proof. From (19) and $\mathbf{y}^* = G(z_0, \mathbf{y}^*)$, one can derive

$$\mathbf{y}^{[N]} - \mathbf{y}^* = -\tau \mathbf{J}(\mathbf{y}^{[N-1]})(\mathbf{y}^{[N-1]} - \mathbf{y}^*) + \tau \mathbf{H}(\mathbf{y}^{[N-1]})\mathbf{A}(\mathbf{F}(\mathbf{y}^{[N-1]}) - \mathbf{F}(\mathbf{y}^*)). \tag{C.1}$$

Taking derivative on both sides of (C.1) with respect to z_0 and using (A.6), it can be shown that

$$\begin{aligned}
\frac{\partial \mathbf{y}^{[N]}}{\partial\ {}_0} - \frac{\partial \mathbf{y}^*}{\partial\ {}_0} = &-\tau\frac{\partial}{\partial\ {}_0}\mathbf{J}(\mathbf{y}^{[N-1]})(\mathbf{y}^{[N-1]} - \mathbf{y}^*) + \tau\frac{\partial}{\partial\ {}_0}\mathbf{H}(\mathbf{y}^{[N-1]})\mathbf{A}(\mathbf{F}(\mathbf{y}^{[N-1]}) - \mathbf{F}(\mathbf{y}^*)) \\
&+ \tau \mathbf{H}(\mathbf{y}^{[N-1]})\mathbf{A}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\right)\frac{\partial \mathbf{y}^*}{\partial\ {}_0}.
\end{aligned} \tag{C.2}$$

By Lemma 3.1 and the mean value theorem, Eq. (C.2) implies that

$$\left\|\frac{\partial \mathbf{y}^{[N]}}{\partial\ {}_0} - \frac{\partial \mathbf{y}^*}{\partial\ {}_0}\right\| \leqslant \tau\left(C_J + C_1 C_H \|A_0\| + \frac{1}{\sqrt{s}}C_2 D_0^2 \|A_0\|\right)\|\mathbf{y}^{[N-1]} - \mathbf{y}^*\|. \tag{C.3}$$

Eq. (35) then follows from Proposition 3.1. Eq. (36) is obtained by making use of (A.6) and (35). □

References

[1] A. Aubry, P. Chartier, Pseudo-symplectic Runge–Kutta methods, BIT 38 (3) (1998) 439–461.

[2] M.A. Austin, P.S. Krishnaprasad, L. Wang, Almost Poisson integration of rigid body systems, J. Comput. Phys. 107 (1) (1993) 105–117.

[3] C. Budd, A. Iserles (Eds.), A Special Issue on Geometric integration: numerical solution of differential equations on manifolds. Vol 357 of Philosophical Transactions of Royal Society of London A. Number 1754, 1999.

[4] J. Candy, W. Rozmus, A symplectic integration algorithm for separable Hamiltonian functions, J. Comput. Phys. 92 (1991) 230–256.

[5] E. Forest, R.D. Ruth, Fourth-order symplectic integration, Physica D 43 (1990) 105–117.

[6] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, Springer, Berlin, New York, 2002.

[7] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonsti. Problems, Springer, Berlin, New York, 1987.

[8] P.S. Krishnaprasad, X. Tan, Cayley transforms in micromagnetics, Physica B 306 (2001) 195–199.

[9] J.E. Marsden, M. West, Discrete mechanics and variational integrators, Acta Numer. (2001) 357–514.

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

[11] J.T. Oden, L.F. Demkowicz, Applied Functional Analysis, CRC Press, Boca Raton, 1996.

[12] R.D. Ruth, A canonical integration technique, IEEE Trans. Nucl. Sci. 30 (4) (1983) 2669–2671.

[13] J.M. Sanz-Serna, Runge–Kutta schemes for Hamiltonian systems, BIT 28 (1988) 877–883.

[14] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, Chapman & Hall, London, New York, 1994.

[15] H.R. Schwarz, Numerical Analysis: A Comprehensive Introduction, Wiley, New York, 1989.

[16] D.R. Smart, Fixed Point Theorems, Cambridge University Press, London, New York, 1974.

[17] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A 150 (5–7) (1990) 262–268.