Almost Symplectic Runge-Kutta Schemes for Hamiltonian Systems

by Xiaobo Tan

CDCSS TR 2004-1
(ISR TR 2004-1)

The Center for Dynamics and Control of Smart Structures (CDCSS) is a joint Harvard University, Boston University, University of Maryland center, supported by the Army Research Office under the ODDR&E MURI97 Program Grant No. DAAG55-97-1-0114 (through Harvard University). This document is a technical report in the CDCSS series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CDCSS/cdcss.html

Almost symplectic Runge-Kutta schemes for Hamiltonian systems

Xiaobo Tan

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

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 we first analyze the preservation of the symplectic structure 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 extensive numerical examples. Numerical comparisons with non-symplectic Runge-Kutta methods and pseudo-symplectic methods are also presented.

AMS: 65L05; 65L06; 65P10

Key words: Geometric integrators; Hamiltonian structure; Symplectic Runge-Kutta methods; Pseudo-symplecticness; 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 (Hairer et al., 2002). For systems evolving on differentiable manifolds (including the important setting of Lie groups), geometric integrators that preserve the manifolds are currently a

Email address: xbtan@umd.edu (Xiaobo Tan).

subject of great interest to theorists and practitioners. See for instance Budd and Iserles (1999). Applications of such techniques are of interest in a variety of physical settings. See for instance Krishnaprasad and Tan (2001) 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 Sanz-Serna and Calvo (1994); Marsden and West (2001) and references therein. 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 (Forest and Ruth, 1990; Yoshida, 1990; Candy and Rozmus, 1991; McLachlan and Atela, 1992). However, for general Hamiltonian systems, the symplectic Runge-Kutta schemes are implicit (Sanz-Serna, 1988). 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 Austin et al. (1993), 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 symplecticness of the numerical flow is established when N iterations are adopted for any $N \geq 1$. 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 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. Choice of the step size, the initial iteration value, and fixed point iteration versus Newton's method are discussed. Numerical comparisons are also conducted with non-symplectic explicit Runge-Kutta methods and with pseudo-symplectic methods proposed in Aubry and Chartier (1998). Note that pseudo-symplectic integrators are explicit and designed to conserve the symplectic structure to a certain order.

The remainder 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 $(p, q) \in \mathbb{R}^d \times Q$ for some integer $d \geq 1$, and Q, the configuration space, is some d-dimensional manifold. In this paper $Q = \mathbb{R}^d$ is assumed for ease of discussion, but the extension of the results to the case of a general Q is straightforward. Let $z \overset{\triangle}{=} \begin{pmatrix} p \\ q \end{pmatrix}$. Then (1) can be rewritten as:

$$\dot{z}(t) = f(z(t)) \overset{\triangle}{=} J\nabla_z H(z(t)), \tag{2}$$

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 (Hairer et al., 1987):

$$\begin{cases} y_i = z_0 + \tau \sum_{j=1}^{s} a_{ij} f(y_j), \; i = 1, \cdots, s \\ z_1 = z_0 + \tau \sum_{i=1}^{s} b_i f(y_i) \end{cases}, \tag{3}$$

where τ is the time step, 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 one time-step flow associated with the algorithm (3), i.e., $z_1 = \Psi_\tau(z_0)$. From Sanz-Serna (1988), the transformation Ψ_τ preserves the symplecticness of the original system (2) if

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

Thus if (4) is satisfied, we have:

$$(\frac{\partial \Psi_\tau}{\partial z_0})^T J (\frac{\partial \Psi_\tau}{\partial z_0}) - J = 0, \tag{5}$$

where "^{T}" 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

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

$b \triangleq (b_1, \cdots, b_s)$, $A_0 \triangleq [a_{ij}]$, and $\mathbf{A} \triangleq A_0 \otimes I_{2d}$, where "$\otimes$" 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

$$\begin{cases} \mathbf{y} = \mathbf{G}(z_0, \mathbf{y}) \triangleq \mathbf{1} \otimes z_0 + \tau \mathbf{A}\mathbf{F}(\mathbf{y}) \\ z_1 = z_0 + \tau b \otimes I_{2d}\mathbf{F}(\mathbf{y}) \end{cases}, \tag{6}$$

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

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 (Hairer et al., 1987). 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.

In this paper $\|\cdot\|$ will be used to denote the 2-norm (or the induced 2-norm) of a vector, a matrix, or a higher rank tensor depending on the context. For an open set Ω, its $\epsilon-$neighborhood, $\mathcal{N}(\Omega,\epsilon)$, is defined as

$$\mathcal{N}(\Omega,\epsilon) \overset{\triangle}{=} \{z \in \mathbb{R}^{2d} : \min_{z_0 \in \bar{\Omega}} \|z - z_0\| \le \epsilon\},$$

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

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

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, $\forall \tau \le \tau_0$, $\forall z_0 \in \Omega$,*

(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

$$\begin{cases} \mathbf{y}^{[n]} = \mathbf{G}(z_0, \mathbf{y}^{[n-1]}) \\ \mathbf{y}^{[0]} = \mathbf{1} \otimes z_0 \end{cases} ; \qquad (7)$$

and

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

Proof. Denote $C_0 \overset{\triangle}{=} \max_{\mathbf{y} \in \mathcal{N}^s(\Omega,\epsilon)} \|\mathbf{F}(\mathbf{y})\|$. Let $\tau_1 = \frac{\epsilon}{C_0\|A_0\|}$ (note that $\|A_0\| = \|\mathbf{A}\|$). Then $\forall \tau \le \tau_1$, $\forall z_0 \in \Omega$, $\mathbf{G}(z_0,\cdot)$ 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 $\tau \le \tau_0 \overset{\triangle}{=} \min\{\tau_1, \tau_2\}$. The rest of the claims then follows from the contraction mapping principle (Smart, 1974). □

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 $N \geq 1$,

$$\begin{cases} \mathbf{y}^{[k]} = \mathbf{G}(z_0, \mathbf{y}^{[k-1]}), \ k = 1, \cdots, N \\ \\ \mathbf{y}^{[0]} = \mathbf{1} \otimes z_0 \\ \\ z_1^{[N]} = z_0 + \tau b \otimes I_{2d} \mathbf{F}(\mathbf{y}^{[N]}) \end{cases} . \tag{8}$$

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

$$\frac{\partial \mathbf{y}^*}{\partial z_0}(z_0) = [I_{2sd} - \tau \mathbf{A} \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*(z_0))]^{-1} (\mathbf{1} \otimes I_{2d}). \tag{9}$$

Similarly z_1 in (6), $\{\mathbf{y}^{[k]}\}_{k=0}^N$ and $z_1^{[N]}$ in (8) (and smooth functions of them) are all continuously differentiable functions of z_0. In the sequel when we write, e.g., $\frac{\partial \mathbf{y}^*}{\partial z_0}$ or $\frac{\partial}{\partial z_0} \mathbf{F}(\mathbf{y}^{[N]})$, *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 one time-step flow associated with the algorithm (8), i.e., $z_1^{[N]} = \Psi_\tau^{[N]}(z_0)$. 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* $\forall \tau \leq \tau_0$, $\forall z_0 \in \Omega$,

$$\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| \leq \frac{D_0(C_1^2 + C_0 C_2 N)\delta^{N+1}}{C_1^2}, \tag{10}$$

$$\|\frac{\partial}{\partial z_0}(\mathbf{F}(\mathbf{y}^{[N]}) - \mathbf{F}(\mathbf{y}^*))\| \leq \frac{D_0(C_1^2 + C_0 C_2 (1+N))\delta^{N+1}}{C_1}, \tag{11}$$

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

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

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

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

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

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* $(\frac{\partial \mathbf{F}}{\partial \mathbf{y}})_{i,j}$ *denotes the* $(i,j)-th$ *component of* $\frac{\partial \mathbf{F}}{\partial \mathbf{y}}$*).*

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 $\forall \tau \le \tau_0$, $\forall z_0 \in \Omega$,*

$$\begin{aligned} \|(\frac{\partial \Psi_\tau^{[N]}(z_0)}{\partial z_0})^T J (\frac{\partial \Psi_\tau^{[N]}(z_0)}{\partial z_0}) - J\| \le & \frac{2\|b\| D_0 D_1 (C_1^2 + C_0 C_2 (1+N))\delta^{N+2}}{\|A_0\| C_1^2} \\ & + (\frac{\|b\| D_0 (C_1^2 + C_0 C_2(1+N))\delta^{N+2}}{\|A_0\| C_1^2})^2 \end{aligned} \tag{15}$$

where

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

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

Proof. Let Ψ_τ be the one time-step flow associated with (6). From (6) and (8),

$$\Lambda^{[N]}(z_0) \overset{\triangle}{=} \Psi_\tau^{[N]}(z_0) - \Psi_\tau(z_0) = \tau b \otimes I_{2d} (\mathbf{F}(\mathbf{y}^{[N]}) - \mathbf{F}(\mathbf{y}^*)).$$

Using Lemma 2.1, one derives

$$\|\frac{\partial \Lambda^{[N]}(z_0)}{\partial z_0}\| \le \frac{\tau \|b\| D_0 (C_1^2 + C_0 C_2 (1+N))\delta^{N+1}}{C_1} \tag{17}$$

Next write

$$\begin{aligned} & \|(\frac{\partial \Psi_\tau^{[N]}(z_0)}{\partial z_0})^T J (\frac{\partial \Psi_\tau^{[N]}(z_0)}{\partial z_0}) - J\| \\ = & \|(\frac{\partial \Lambda^{[N]}(z_0)}{\partial z_0} + \frac{\partial \Psi_\tau(z_0)}{\partial z_0})^T J (\frac{\partial \Lambda^{[N]}(z_0)}{\partial z_0} + \frac{\partial \Psi_\tau(z_0)}{\partial z_0}) - J\| \\ \le & \|(\frac{\partial \Lambda^{[N]}(z_0)}{\partial z_0})^T J (\frac{\partial \Lambda^{[N]}(z_0)}{\partial z_0})\| + 2\|(\frac{\partial \Lambda^{[N]}(z_0)}{\partial z_0})^T J (\frac{\partial \Psi_\tau(z_0)}{\partial z_0})\| + \|(\frac{\partial \Psi_\tau(z_0)}{\partial z_0})^T J (\frac{\partial \Psi_\tau(z_0)}{\partial z_0}) - J\|, \end{aligned}$$

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

$$\|\frac{\partial \Psi_\tau(z_0)}{\partial z_0}\| = \|I_{2d} + \tau b \otimes I_{2d} \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*) \frac{\partial \mathbf{y}^*}{\partial z_0}\| \le D_1. \tag{18}$$

□

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 $(c_1 + c_2 N)\delta^{N+2}$ 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

$$\mathbf{y}^{[n]} = \tilde{\mathbf{G}}(z_0, \mathbf{y}^{[n-1]}) \triangleq \mathbf{y}^{[n-1]} - [I_{2sd} - \tau \mathbf{A}\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[n-1]})]^{-1}(\mathbf{y}^{[n-1]} - \mathbf{1} \otimes z_0 - \tau \mathbf{A}\mathbf{F}(\mathbf{y}^{[n-1]})). \tag{19}$$

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 (Schwarz, 1989). 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, $\forall \tau \leq \tau_0$, $\forall z_0 \in \Omega$,*

(1) $\tilde{\mathbf{G}}(z_0, \cdot)$ 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

$$\begin{cases} \mathbf{y}^{[n]} = \tilde{\mathbf{G}}(z_0, \mathbf{y}^{[n-1]}) \\ \mathbf{y}^{[0]} = \mathbf{1} \otimes z_0 \end{cases} ; \tag{20}$$

and

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

Proof. Through algebraic manipulations, $\tilde{\mathbf{G}}(z_0, \mathbf{y})$ can be rewritten as

$$\tilde{\mathbf{G}}(z_0, \mathbf{y}) = \mathbf{1} \otimes z_0 + \tau[I_{2sd} - \tau \mathbf{A}\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y})]^{-1}\mathbf{A}(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y})(\mathbf{1} \otimes z_0 - \mathbf{y}) + \mathbf{F}(\mathbf{y})). \tag{21}$$

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

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

$$E_1 \triangleq \max_{\mathbf{y} \in \mathcal{N}^s(\Omega,\epsilon), z_0 \in \Omega} \|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y})(\mathbf{1} \otimes z_0 - \mathbf{y}) + \mathbf{F}(\mathbf{y})\|, \tag{23}$$

and let $\tau_2 > 0$ be such that $\tau_2 E_0 E_1 \|A_0\| < 1$. Then it can be verified that if $\tau \leq \min\{\tau_1, \tau_2\}$, $\tilde{\mathbf{G}}(z_0, \cdot)$ maps $\mathcal{N}^s(\Omega, \epsilon)$ into itself.

The next goal is to establish that $\tilde{\mathbf{G}}(z_0, \cdot)$ 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 $\frac{\partial^2 \mathbf{F}}{\partial \mathbf{y}^2}$ involved, for $\eta \in \mathbb{R}^{2sd}$, one calculates

$$\frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{y}}(z_0, \mathbf{y}) \cdot \eta = -\tau \mathbf{H}(\mathbf{y}) \mathbf{A}(\frac{\partial^2 \mathbf{F}}{\partial \mathbf{y}^2}(\mathbf{y}) \cdot \eta) \mathbf{H}(\mathbf{y})[\mathbf{y} - \mathbf{1} \otimes z_0 - \tau A \mathbf{F}(\mathbf{y})], \tag{24}$$

where "·" denotes the action of a second-rank or third-rank tensor on a vector, and

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

Denote

$$E_2 \triangleq \max_{\mathbf{y} \in \mathcal{N}^s(\Omega,\epsilon)} \|\frac{\partial^2 \mathbf{F}}{\partial \mathbf{y}^2}(\mathbf{y})\|, \tag{26}$$

$$E_3 \triangleq \max_{\mathbf{y} \in \mathcal{N}^s(\Omega,\epsilon), z_0 \in \Omega, \tau \leq \tau_1} \|\mathbf{y} - \mathbf{1} \otimes z_0 - \tau \mathbf{A} \mathbf{F}(\mathbf{y})\|, \tag{27}$$

and pick $\tau_3 > 0$ such that $\tau_3 E_0^2 E_2 E_3 \|A_0\| < 1$. Then when $\tau \leq \min\{\tau_1, \tau_2, \tau_3\}$, $\tilde{\mathbf{G}}(z_0, \cdot)$ is a contraction mapping and hence (20) converges to a (unique) fixed point, which is the solution to the first equation in (6).

Since $\frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{y}}(z_0, \mathbf{y}^*) = 0$, the convergence rate of (20) is quadratic, as is standard for Newton's method (Schwarz, 1989):

$$\|\mathbf{y}^{[n]} - \mathbf{y}^*\| \leq K \|\mathbf{y}^{[n-1]} - \mathbf{y}^*\|^2 \leq K^{2^n - 1} \|\mathbf{y}^{[0]} - \mathbf{y}^*\|^{2^n}, \tag{28}$$

where

$$K \triangleq \max_{\mathbf{y} \in \mathcal{N}^s(\Omega,\epsilon), z_0 \in \Omega, \tau \leq \tau_1} \|\frac{\partial^2 \tilde{\mathbf{G}}}{\partial \mathbf{y}^2}(z_0, \mathbf{y})\|. \tag{29}$$

It's easy to see that $\frac{\partial^2 \tilde{\mathbf{G}}}{\partial \mathbf{y}^2}(z_0, \mathbf{y})$ contains a factor of τ. On the other hand, $\|\mathbf{y}^{[0]} - \mathbf{y}^*\| \leq \tau C_0 \|A_0\|$, where C_0 is as defined in Lemma 2.1. Therefore there exists $\tau_4 > 0$ such that when $\tau \leq \tau_4$, $K \|\mathbf{y}^* - \mathbf{y}^{[0]}\| < 1$. 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 $N \geq 1$,

$$\begin{cases} \mathbf{y}^{[k]} = \tilde{\mathbf{G}}(z_0, \mathbf{y}^{[k-1]}),\ k = 1, \cdots, N \\ \mathbf{y}^{[0]} = \mathbf{1} \otimes z_0 \\ z_1^{[N]} = z_0 + \tau b \otimes I_{2d}\mathbf{F}(\mathbf{y}^{[N]}) \end{cases} . \tag{30}$$

Denote by $\tilde{\Psi}_\tau^{[N]}$ the one time-step flow associated with the algorithm (30). 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 $\mathcal{N}(\Omega, \epsilon)$. 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 $\forall \tau \leq \tau_0$, $\forall z_0 \in \Omega$,*

$$\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0}\| \leq C_y \triangleq \sqrt{s}(1 + \frac{E_0}{1-\gamma_0}), \tag{31}$$

$$\|\frac{\partial}{\partial z_0}\mathbf{H}(\mathbf{y}^{[N]})\| \leq C_H \triangleq \frac{\gamma_0 C_y}{E_3}, \tag{32}$$

$$\|\frac{\partial}{\partial z_0}\mathbf{J}(\mathbf{y}^{[N]})\| \leq C_J \triangleq \frac{\|A_0\|(C_1\gamma_0 + E_0E_2E_3)C_y}{E_3}, \tag{33}$$

where $\gamma_0 \triangleq \tau_0 E_0^2 E_2 E_3 \|A_0\|$; C_1 is as defined in (13); E_1, E_2 are as defined in (23), (26); and E_0 and E_3 are as defined in (22), (27) 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 $\forall \tau \leq \tau_0$, $\forall z_0 \in \Omega$,*

$$\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| \leq D_y \delta^{2^{N-1}}, \tag{34}$$

$$\|\frac{\partial}{\partial z_0}\mathbf{F}(\mathbf{y}^{[N]}) - \frac{\partial}{\partial z_0}\mathbf{F}(\mathbf{y}^*)\| \leq C_1 D_y \delta^{2^{N-1}} + \frac{C_2 D_0}{K}\delta^{2^N}, \tag{35}$$

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 (29), 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 one time-step flow associated with (30). Then $\forall \tau \leq \tau_0$, $\forall z_0 \in \Omega$,*

$$\|(\frac{\partial \tilde{\Psi}_\tau^{[N]}(z_0)}{\partial z_0})^T J (\frac{\partial \tilde{\Psi}_\tau^{[N]}(z_0)}{\partial z_0}) - J\| \leq 2\tau D_1 \|b\| (C_1 D_y \delta^{2^{N-1}} + \frac{C_2 D_0}{K} \delta^{2^N}) + (\tau \|b\| (C_1 D_y \delta^{2^{N-1}} + \frac{C_2 D_0}{K} \delta^{2^N}))^2 \quad (36)$$

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 (30) on symplectic structure conservation have been characterized in Theorem 2.1 and Theorem 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 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 (symplecticness) 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 symplecticness and the computational complexity. For illustrational purposes, the methods listed in Table 1 will be compared in the numerical problems. For a definition of pseudo-symplecticness order, we refer to Aubry and Chartier (1998). 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 Aubry and Chartier (1998). 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 area-preservation. The ellipse shown in Fig. 1, with semi-major axis = 1.8 and semi-minor axis = 1.2, represents the set of initial conditions for the nonlinear pendulum problem. Numerical solutions after one time-step under different methods are compared with the exact solution in Fig. 2, where the time

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 (Hairer et al., 2002)	4	symplectic	2
PS63	Pseudo-symp. method (Aubry and Chartier, 1998)	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 Conditions
Nonlinear pendulum	$\frac{p^2}{2} - \cos(q)$	1.6, 0.8, 0.2	See the text
Linear pendulum	$\frac{1}{2}(p^2 + q^2)$	0.5	$(2,2)^T$
Kepler problem	$\frac{1}{2}(p_1^2 + p_2^2) - \frac{1}{\sqrt{q_1^2+q_2^2}}$	0.1	$(0,2,0.4,0)^T$
Bead on a wire	$\frac{p^2}{2(1+U'(q)^2)} + U(q)$ with $U(q) = 0.1(q(q-2))^2 + 0.008q^3$	$\frac{1}{6}$	$(0.49,0)^T$
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	$(0,1.689,0.2,2.5,0,0)^T$

step $\tau = 1.6$, 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 symplecticity of the exact flow, the area enclosed by the exact solutions at $t = 1.6$ is equal to that enclosed by the initial curve. 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.

To provide a quantitative measure of area preservation, we have picked 10^4 points on the ellipse (as initial conditions).

The initial area $\mathcal{A}_0$ at time $t = 0$ is approximated by the sum of areas of 10^4 triangles formed by the picked points and the origin (see Fig. 1 for illustration, where 8 points are used). The final area $\mathcal{A}_1$ at $t = 1.6$ is calculated similarly, using the current 10^4 solution points. The (normalized) area error is then defined as $\frac{|\mathcal{A}_1 - \mathcal{A}_0|}{\mathcal{A}_0}$.

One goal of this paper is to provide insight into the choice of fixed-point iteration versus Newton's method. From Theorem 2.1 and Theorem 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 Fig. 3 and Fig. 4. Fig. 3 shows the decrease of area error with the number of fixed-point iterations, where the underlying algorithm used 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. In Fig. 4, the area error stops decreasing after 4 iterations, and it stays around 0.5×10^{-8}. This is not due to the limitation of machine accuracy. It is considered to arise from approximating the area based on 10^4 points. Indeed, we have observed that the error stops decreasing around 10^{-6} if 10^3 points are used in computing the area.

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 actual computation time is considered. In terms of N, the computation 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 computation overhead and the computation 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.

Two other step sizes $\tau = 0.8$, $\tau = 0.2$ are used to integrate the nonlinear pendulum equation while the final time is kept the same, i.e., $t = 1.6$. Therefore the time 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 error 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 time steps which, to keep the total CPU time the same, leads to a small number N of iterations at each time 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 error down to the order of 10^{-8} (the achievable limit as explained earlier), 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 error 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 (Fig. 8 and Fig. 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.

From (28) 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. Again when looking at the figure, one should keep in mind that 0.5×10^{-8} is the achievable area-error limit as a result of our area-approximation method, and hence he or she should not be confused by the somewhat misleading slopes of the last segments of the curves.

4.2 Other problems

Fig. 12 shows the trajectories of the linear pendulum in the phase space under Gauss4/FixedPt and Gauss4/Newton for 5×10^4 time steps (the data were down-sampled by 20 to reduce the file size). For the fixed-point iteration method, the energy decays to zero if the iteration number $N = 3$. The energy decay rate is significantly reduced when N is increased to 5, and with $N = 8$, the trajectory almost stays on the circular orbit. Newton's method, on the other hand, gives rise to the exact solution (up to the machine precision) in one iteration since the system is linear.

The numerical solutions of the Kepler problem (q_1 and q_2 components) are plotted in Fig. 13 (after down-sampling by 20). For comparison, the exact orbit is also shown. The total number of time steps is 2×10^4. It can be observed that when $N = 6$, the solution with Gauss4/FixedPt follows a precession motion of the elliptical orbit. This is also true for Gauss4/Newton ($N = 2$). Such "precession" effect is typical when integrating the Kepler problem with a symplectic scheme (Hairer et al., 2002). Note that PS63 also demonstrates the similar behavior with a slower precession rate. For Gauss4/FixedPt with $N = 2, 4$ and RK4, the solutions distort the ellipse. The angular momentum is also 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.

Fig. 15 and Fig. 16 show the evolution of error in the Hamiltonian for the bead-on-a-wire problem and the galactic dynamics problem. Table 4 and Table 5 list the CPU time used by different algorithms.

Table 3

CPU time used in solving the Kepler problem.

	Gauss4/FixedPt					Gauss4/Newton				
Method	N	2	4	6	8	N	2	3	PS63	RK4
Time (sec.)		49.9	70.6	89.3	108.2		73.0	91.0	55.0	44.7

Table 4

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

	Gauss4/FixedPt				Gauss4/Newton					
Method	N	3	5	7	N	1	2	3	PS63	RK4
Time (sec.)		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.

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

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 $\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\|$) 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 Proposition 2.1 and Proposition 3.1 may be used to find the step size τ_0 (below which the scheme is 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 Aubry and Chartier (1998), the design of a pseudo-symplectic method (in particular, of order p and of pseudo-symplecticness order $2p$ (Aubry and Chartier, 1998)) beyond order (3,6) is very complicated. This will hinder the use of pseudo-symplectic methods in very long time simulation of Hamiltonian systems.

Acknowledgement.

The author would like to thank Professor P. S. Krishnaprasad for numerous discussions and valuable suggestions on this work. 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.

References

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

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

Budd, C., Iserles, A. (Eds.), 1999. 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.

Candy, J., Rozmus, W., 1991. A symplectic integration algorithm for separable Hamiltonian functions. Journal of Comput. Phys. 92, 230–256.

Forest, E., Ruth, R. D., 1990. Fourth-order symplectic integration. Phys. D 43, 105–117.

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

Hairer, E., Nørsett, S. P., Wanner, G., 1987. Solving Ordinary Differential Equations I: Nonstiff Problems. Springer-Verlag.

Krishnaprasad, P. S., Tan, X., 2001. Cayley transforms in micromagnetics. Physica B 306, 195–199.

Marsden, J. E., West, M., 2001. Discrete mechanics and variational integrators. Acta Numerica , 357–514.

McLachlan, R. I., Atela, P., 1992. The accuracy of symplectic integrators. Nonlinearity 5, 541–562.

Sanz-Serna, J. M., 1988. Runge-Kutta schemes for Hamiltonian systems. BIT 28, 877–883.

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

Schwarz, H. R., 1989. Numerical Analysis: A Comprehensive Introduction. Wiley.

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

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

A Proof of Lemma 2.1

Proof. From (6) and (8),

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

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

$$\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0} = \tau \mathbf{A}[\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]})(\frac{\partial \mathbf{y}^{[N-1]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}) + (\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*))\frac{\partial \mathbf{y}^*}{\partial z_0}]. \tag{A.2}$$

Eq. (A.2) implies

$$\begin{aligned} \|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| &\leq \tau \|A_0\| \, \|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]})\| \, \|\frac{\partial \mathbf{y}^{[N-1]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| + \tau \|A_0\| \, \|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\| \, \|\frac{\partial \mathbf{y}^*}{\partial z_0}\| \\ &\leq \tau C_1 \|A_0\| \, \|\frac{\partial \mathbf{y}^{[N-1]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| + \tau D_0 \|A_0\| \, \|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\|. \end{aligned} \tag{A.3}$$

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

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

which leads to

$$\begin{aligned} \|\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\| &\leq C_2 \|\mathbf{y}^{[N-1]} - \mathbf{y}^*\| \\ &\leq C_2 \delta^{N-1} \|\mathbf{y}^{[0]} - \mathbf{y}^*\| \text{ (from Proposition 2.1)} \\ &\leq \tau C_0 C_2 \|A_0\| \delta^{N-1} \| \text{ (since } \mathbf{y}^* - \mathbf{y}^{[0]} = \tau \mathbf{A}\mathbf{F}(\mathbf{y}^*)). \end{aligned} \tag{A.4}$$

Plugging (A.4) into (A.3) and performing recursions, one has

$$\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| \leq \delta^N \|\frac{\partial \mathbf{y}^{[0]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| + \frac{C_0 C_2 D_0 N \delta^{N+1}}{C_1^2}.$$

Eq. (10) is then proved by noting

$$\|\frac{\partial \mathbf{y}^{[0]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| = \|\tau \mathbf{A}\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*)\frac{\partial \mathbf{y}^*}{\partial z_0}\| \leq \tau C_1 D_0 \|A_0\|.$$

To show (11), write

$$\frac{\partial}{\partial z_0}(\mathbf{F}(\mathbf{y}^{[N]}) - \mathbf{F}(\mathbf{y}^*)) = \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N]})(\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}) + (\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*))\frac{\partial \mathbf{y}^*}{\partial z_0}, \tag{A.6}$$

and then use (10) and (A.4). □

B Proof of Lemma 3.1

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

$$\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} = \frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{y}}(z_0, \mathbf{y}^{[N-1]})\frac{\partial \mathbf{y}^{[N-1]}}{\partial z_0} + \frac{\partial \tilde{\mathbf{G}}}{\partial z_0}(z_0, \mathbf{y}^{[N-1]}). \tag{B.1}$$

From $\|\frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{y}}(z_0, \mathbf{y}^{[N-1]})\| \leq \tau E_0^2 E_2 E_3 \|A_0\|$ (recall (24)) and $\|\frac{\partial \tilde{\mathbf{G}}}{\partial z_0}(z_0, \mathbf{y}^{[N-1]})\| = \|\mathbf{H}(\mathbf{y}^{[N-1]})\mathbf{1} \otimes I_{2d}\| \leq \sqrt{s}E_0$, one gets

$$\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0}\| \leq \tau E_0^2 E_2 E_3 \|A_0\| \ \|\frac{\partial \mathbf{y}^{[N-1]}}{\partial z_0}\| + \sqrt{s}E_0.$$

Since $\|\frac{\partial \mathbf{y}^{[0]}}{\partial z_0}\| = \sqrt{s}$,

$$\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0}\| \leq \sqrt{s}(\gamma^N + \frac{E_0(1-\gamma^N)}{1-\gamma}),$$

where $\gamma \triangleq \tau E_0^2 E_2 E_3 \|A_0\|$. Eq. (31) then follows from $0 < \gamma \leq \gamma_0 < 1$.

Eq. (32) can be shown by writing

$$\frac{\partial}{\partial z_0}\mathbf{H}(\mathbf{y}^{[N]}) = \frac{\partial \mathbf{H}}{\partial \mathbf{y}}(\mathbf{y}^{[N]})\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} = \tau \mathbf{H}(\mathbf{y}^{[N]})\mathbf{A}\frac{\partial^2 \mathbf{F}}{\partial \mathbf{y}^2}(\mathbf{y}^{[N]})\mathbf{H}(\mathbf{y}^{[N]})\frac{\partial \mathbf{y}^{[N]}}{\partial z_0}$$

and then using (31).

Finally to show (33), note that

$$\frac{\partial}{\partial z_0}\mathbf{J}(\mathbf{y}^{[N]}) = \frac{\partial}{\partial z_0}\mathbf{H}(\mathbf{y}^{[N]})\mathbf{A}\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N]}) + \mathbf{H}(\mathbf{y}^{[N]})\mathbf{A}\frac{\partial^2 \mathbf{F}}{\partial \mathbf{y}^2}(\mathbf{y}^{[N]})\frac{\partial \mathbf{y}^{[N]}}{\partial z_0},$$

and then use (31) and (32). □

C Proof of Lemma 3.2

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

$$\begin{aligned}\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0} = &-\tau \frac{\partial}{\partial z_0}\mathbf{J}(\mathbf{y}^{[N-1]})(\mathbf{y}^{[N-1]} - \mathbf{y}^*) + \tau \frac{\partial}{\partial z_0}\mathbf{H}(\mathbf{y}^{[N-1]})\mathbf{A}(\mathbf{F}(\mathbf{y}^{[N-1]}) - \mathbf{F}(\mathbf{y}^*)) \\ &+\tau \mathbf{H}(\mathbf{y}^{[N-1]})\mathbf{A}(\frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^{[N-1]}) - \frac{\partial \mathbf{F}}{\partial \mathbf{y}}(\mathbf{y}^*))\frac{\partial \mathbf{y}^*}{\partial z_0}.\end{aligned} \tag{C.2}$$

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

$$\|\frac{\partial \mathbf{y}^{[N]}}{\partial z_0} - \frac{\partial \mathbf{y}^*}{\partial z_0}\| \leq \tau(C_J + C_1 C_H \|A_0\| + \frac{1}{\sqrt{s}} C_2 D_0^2 \|A_0\|)\|\mathbf{y}^{[N-1]} - \mathbf{y}^*\|.$$

Eq. (34) then follows from Proposition 3.1. Eq. (35) is obtained by making use of (A.6) and (34). □

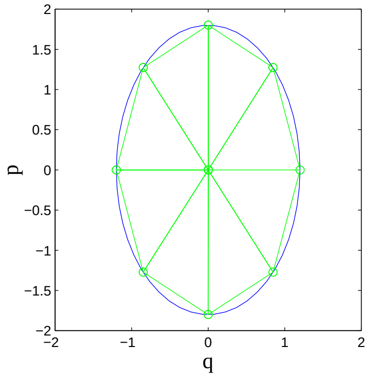

Fig. 1. Initial conditions for the nonlinear pendulum problem and the schematic of approximating the enclosed area with a finite number of triangles.

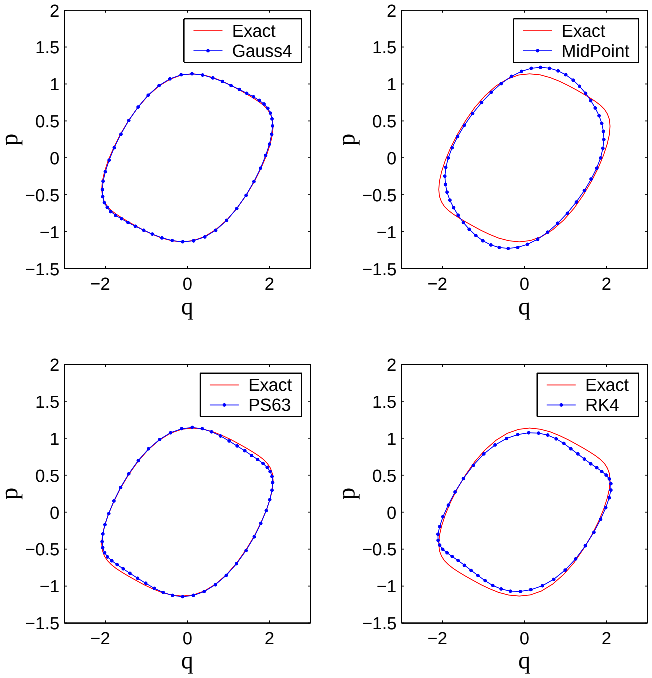

Fig. 2. Comparison of numerical solutions with the exact one at $t = 1.6$ ($\tau = 1.6$) for the nonlinear pendulum problem.

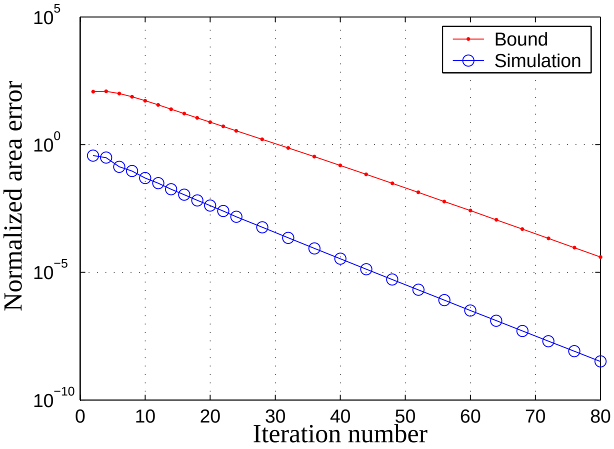

Fig. 3. Decrease of the area error *vs* the number N of iterations with fixed-point iteration computed for the nonlinear pendulum problem. MidPoint is used with $\tau = 1.6$ and the number of time steps is one.

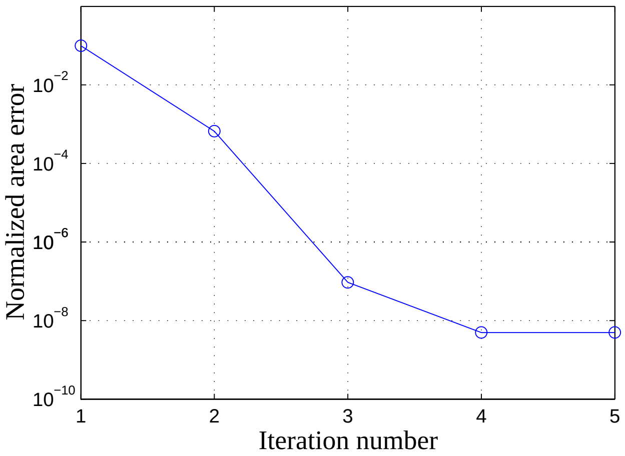

Fig. 4. Decrease of the area error *vs* the number N of iterations with Newton's method computed for the nonlinear pendulum problem.. MidPoint is used with $\tau = 1.6$ and the number of time steps is one.

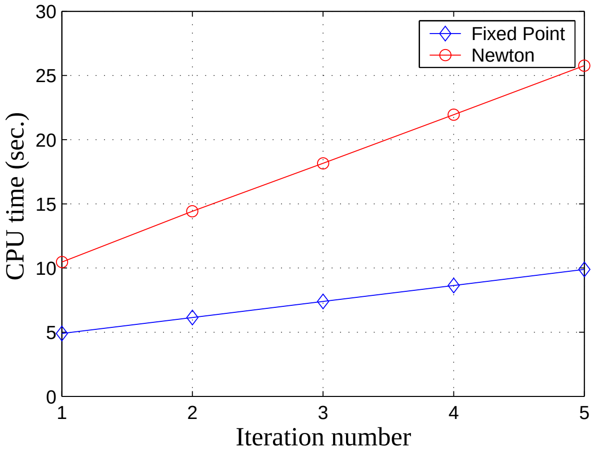

Fig. 5. Comparison of the computation time (for one time-step) *vs* the number N of iterations for fixed-point iteration and Newton's method. The nonlinear pendulum problem is computed and MidPoint used with $\tau = 1.6$.

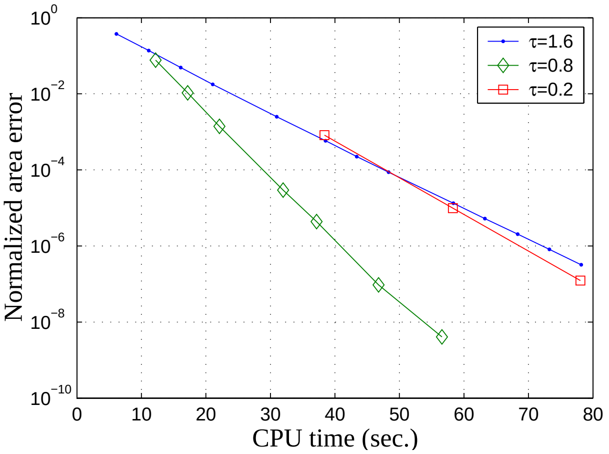

Fig. 6. Work-precision diagrams for the nonlinear pendulum problem under the fixed-point iteration scheme with different step sizes. Final time $t = 1.6$ fixed. Underlying algorithm: MidPoint.

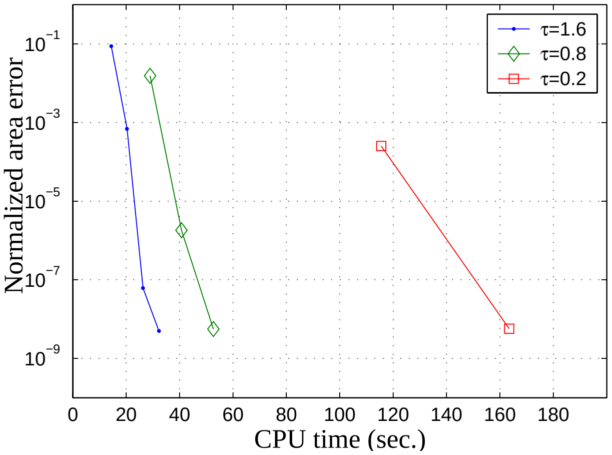

Fig. 7. Work-precision diagrams for the nonlinear pendulum problem under Newton's method-based scheme with different step sizes. Final time $t = 1.6$ fixed. Underlying algorithm: MidPoint.

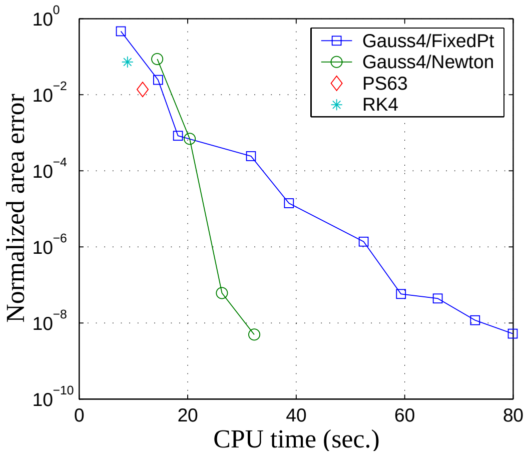

Fig. 8. Comparison of work-precision diagrams for the nonlinear pendulum problem under different schemes ($\tau = 1.6$). Final time $t = 1.6$.

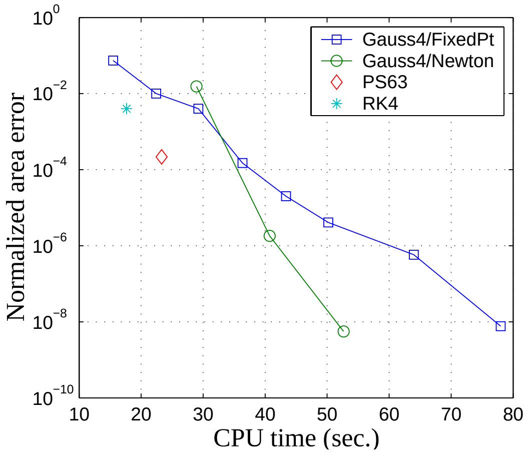

Fig. 9. Comparison of work-precision diagrams for the nonlinear pendulum problem under different schemes ($\tau = 0.8$). Final time $t = 1.6$.

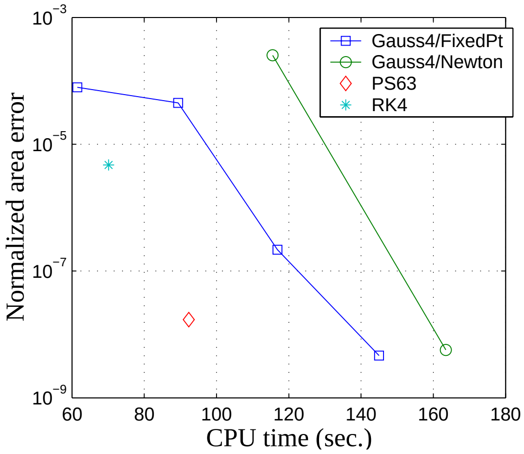

Fig. 10. Comparison of work-precision diagrams for the nonlinear pendulum problems under different schemes ($\tau = 0.2$). Final time $t = 1.6$.

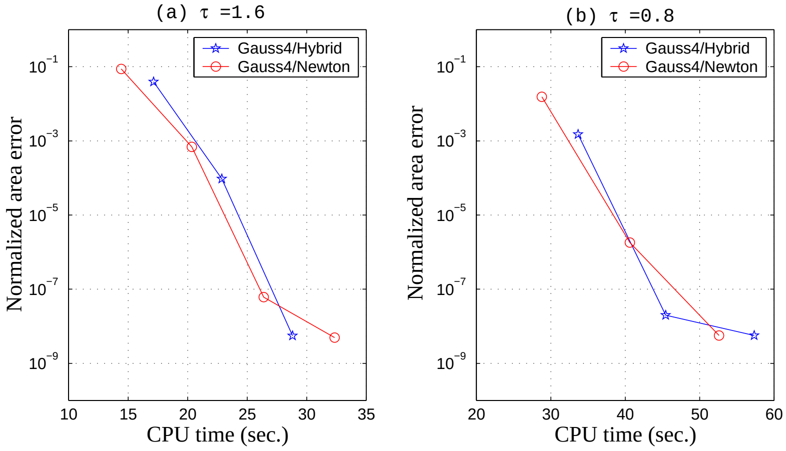

Fig. 11. Comparison of work-precision diagrams for the nonlinear pendulum problem under the hybrid scheme and Newton's method, where the underlying algorithm is Gauss4. Gauss4/Hybrid: run fixed point iteration once and then run Newton's method. (a) $\tau = 1.6$; (b) $\tau = 0.8$. Final time $t = 1.6$ for both (a) and (b).

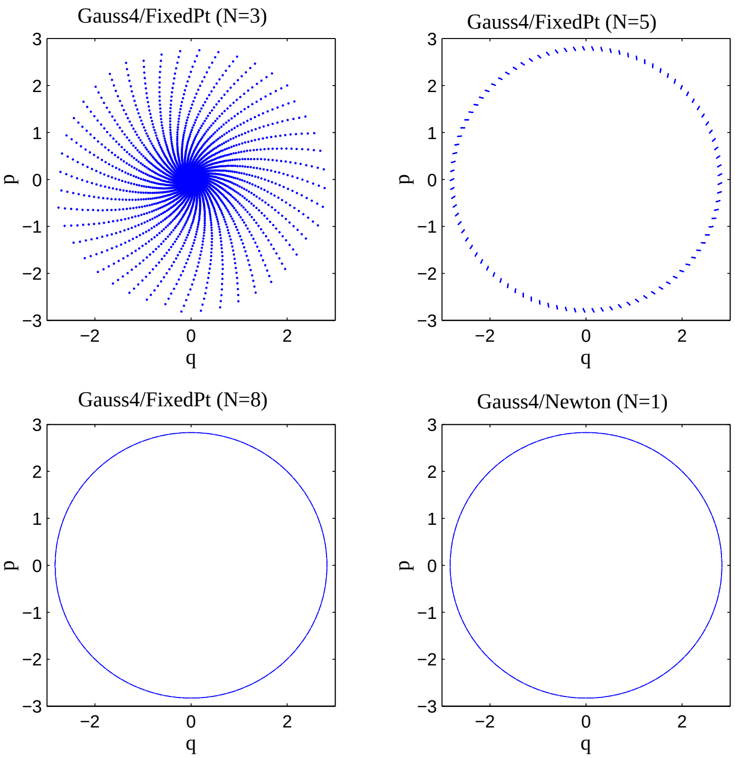

Fig. 12. Trajectories of the linear pendulum in the phase space under Gauss4/FixedPt and Gauss4/Newton.

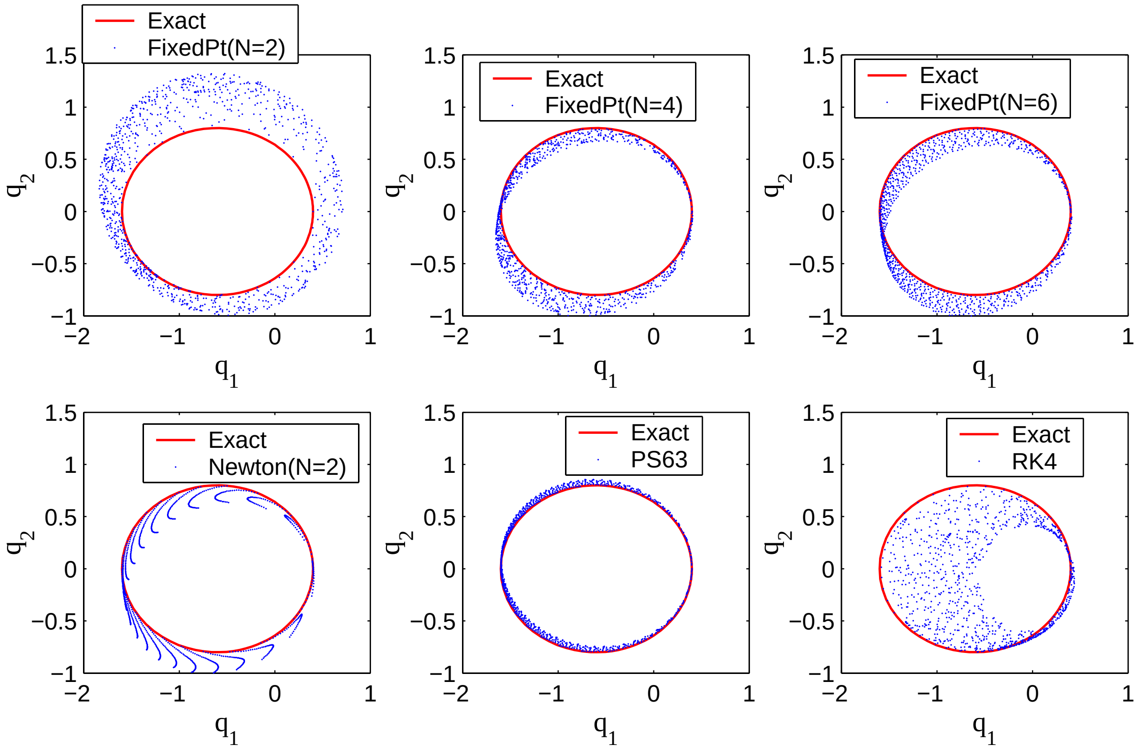

Fig. 13. Exact and numerical solutions of the Kepler problem. The underlying algorithm for FixedPt and Newton was Gauss4.

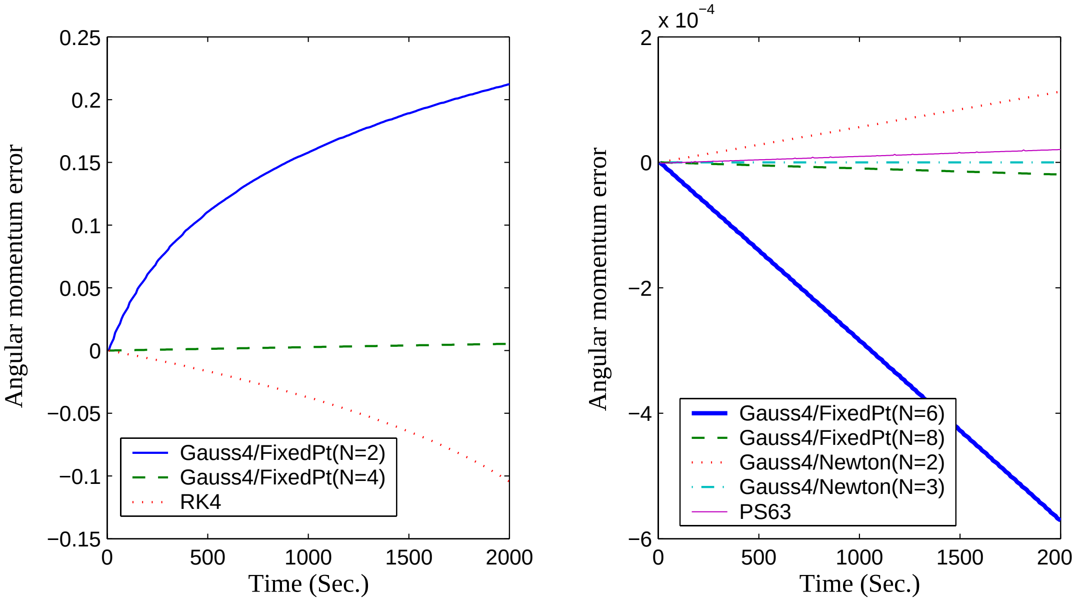

Fig. 14. Comparison of the angular momentum error for the Kepler problem.

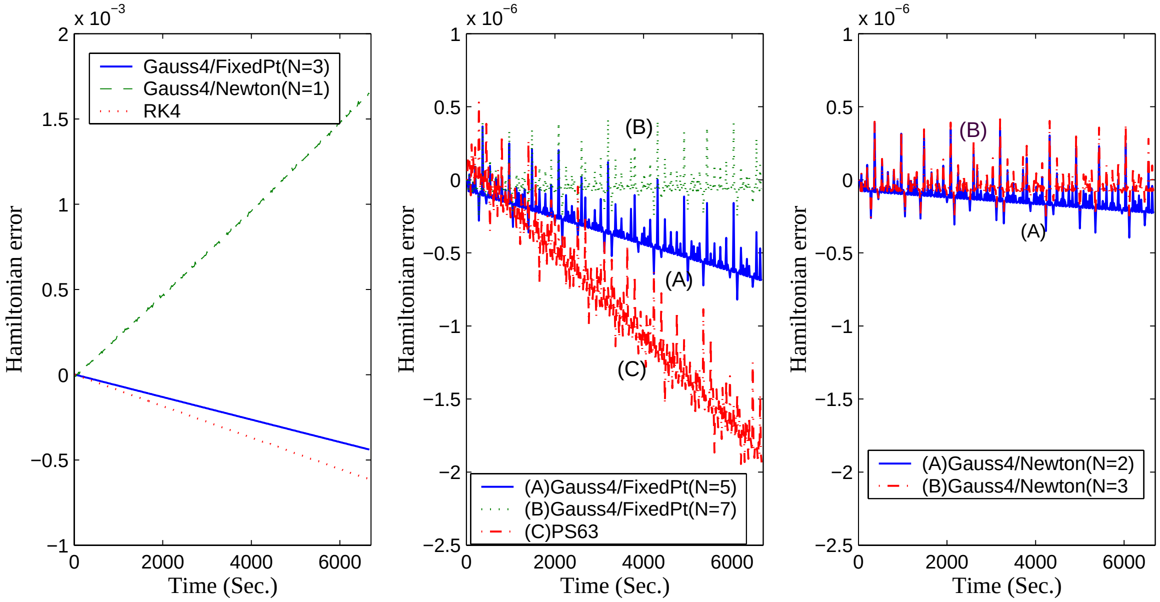

Fig. 15. Comparison of the Hamiltonian error for the bead-on-a-wire problem.

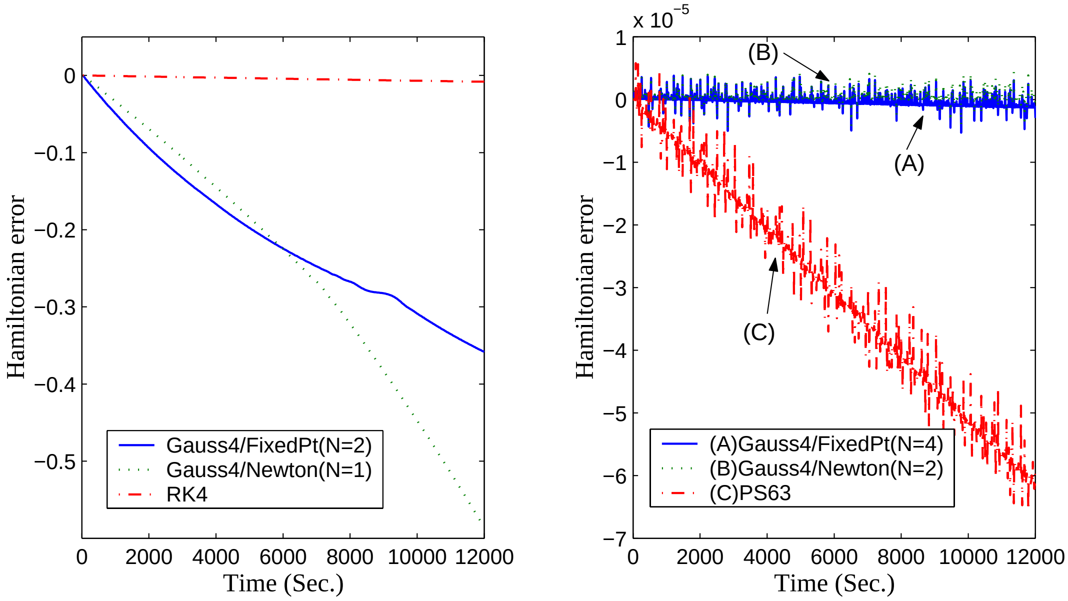

Fig. 16. Comparison of the Hamiltonian error for the galactic dynamics problem.