Journal of Computational Physics **160,** 88–116 (2000)
doi:10.1006/jcph.2000.6427, available online at http://www.idealibrary.com on IDEAL®

Inherently Energy Conserving Time Finite Elements for Classical Mechanics

P. Betsch* and P. Steinmann†

Department of Mechanical Engineering, University of Kaiserslautern, Postfach 3049, 67653 Kaiserslautern, Germany
E-mail: *pbetsch@rhrk.uni-kl.de and †ps@rhrk.uni-kl.de

Received October 27, 1998; revised November 24, 1999

In this paper, we develop a finite element method for the temporal discretization of the equations of motion. The continuous Galerkin method is based upon a weighted-residual statement of Hamilton's canonical equations. We show that the proposed finite element formulation is energy conserving in a natural sense. A family of implicit one-step algorithms is generated by specifying the polynomial approximation in conjunction with the quadrature formula used for the evaluation of time integrals. The numerical implementation of linear, quadratic, and cubic time finite elements is treated in detail for the model problem of a circular pendulum. In addition to that, concerning dynamical systems with several degrees of freedom, we address the design of nonstandard quadrature rules which retain the energy conservation property. Our numerical investigations assess the effect of numerical quadrature in time on the accuracy and energy conservation property of the time-stepping schemes.

Key Words: Petrov–Galerkin method; finite element method; initial value problems; classical mechanics; Hamilton's equations.

1. INTRODUCTION

In this paper we develop a finite element formulation for the temporal discretization of the equations of motion. We restrict ourselves to holonomic dynamical systems formulated in terms of independent generalized coordinates. The newly developed time finite element formulation is based upon the temporal discretization of Hamilon's canonical equations by means of the continuous Galerkin (cG) method.

Following the terminology of Eriksson *et al.* [7] the term "cG(k) method" refers to trial functions consisting of continuous piecewise polynomials of degree k and test functions consisting of discontinuous piecewise polynomials of degree $k-1$. The cG method has apparently been introduced by Hulme [16] for the numerical solution of systems of first-order ordinary differential equations.

0021-9991/00 $35.00

We show that the cG method in conjunction with Hamilton's equations is inherently energy conserving. That is, provided that the time integrals appearing in the time finite element formulation are calculated exactly, the resulting time-stepping scheme is exactly energy conserving. Of course, exact calculation of the time integrals is rarely feasible. Therefore, we investigate the effect of standard quadrature rules on algorithmic energy conservation. Moreover, we show that it is possible to maintain exact algorithmic energy conservation by the design of nonstandard quadrature formulas. Concerning the application of quadrature rules it is interesting to note that the cG(k) method yields time-stepping schemes that coincide with k-stage Gauss Runge–Kutta methods if k-point Gaussian quadrature formulas are used. This was already shown in Hulme [16].

Nonlinear elastodynamics is but one field where algorithmic energy conservation appears to be a desirable property of time-stepping schemes, especially from the viewpoint of numerical stability (see, e.g., Hughes [14] and Gonzalez and Simo [11]). In this context Simo and Tarnow [20] and Crisfield and Shi [6] have shown that the lack of algorithmic energy conservation can lead to dramatic blowup behavior. As a remedy they render the midpoint rule energy preserving by employing a modified stress calculation. Another approach relies on the application of Lagrange multipliers for the algorithmic enforcement of the energy constraint (see, e.g., Hughes *et al.* [15] or Kuhl and Ramm [17]). Alternatively, the energy constraint equation may be used to solve for an additional scalar variable which is introduced into the time-stepping algorithm in order to fulfill energy conservation (see, e.g., LaBudde and Greenspan [18] and Simo *et al.* [21]). Furthermore, the application of "discrete gradient" methods to Hamiltonian systems yields energy conserving time-stepping schemes (see Gonzalez [10] and McLachlan *et al.* [19] and references therein).

The variational formulation used herein can be related to Hamilton's law of varying action as well as Hamilton's principle (see Remark 3.1 below), which have previously been the starting point for the development of alternative time-stepping schemes. For example, Hamilton's principle in conjunction with the Ritz method can be employed for the temporal discretization of dynamical systems with specified end-point conditions (see, e.g., Gillilan and Wilson [8] or the early work of Argyris and Scharpf [1]). Another approach relies on the introduction of a discrete variational principle which can be used to obtain the associated discrete Euler–Lagrange equations (see Wendlandt and Marsden [23] and references therein). Hamilton's law of varying action is the starting point of the discretization method advocated by Bailey [3], where global polynomial approximations of the displacements are applied.

An outline of the remainder of the paper is as follows. In section 2 we give a brief summary of the Hamiltonian formulation of the equations of motion needed for the subsequent developments. Section 3 contains the temporal discretization of Hamilton's equations by means of the continuous Galerkin method. Making use of the subparametric finite element concept along with Lagrangean shape functions we arrive at the general finite element formulation. In addition to that, the algorithmic conservation properties are investigated, namely (i) conservation of total energy and (ii) conservation of generalized momenta corresponding to cyclic coordinates. Section 4 is devoted to computational aspects in the realm of one-dimensional motion. In particular, we give a detailed account of the numerical implementation of the method related to $k = 1, 2, 3$. In this connection, numerical simulations are given to examine algorithmic energy conservation as well as the error in the generalized displacements/momenta. Computational aspects pertaining to systems with several degrees of freedom are addressed in Section 5. The corresponding numerical example deals with a

planar double pendulum. Conclusions are drawn in Section 6. The connection with often applied Gauss Runge–Kutta methods is verified in the Appendix.

2. HAMILTONIAN FORMULATION OF THE EQUATIONS OF MOTION

We summarize below some results of classical mechanics which will be needed for the subsequent development of the time-stepping algorithms. In particular, we concentrate on the Hamiltonian formulation of the equations of motion which are the starting point for the Galerkin approximations developed below. We refer to the books of Goldstein [9] and Arnold [2] for a more detailed account of the subject.

Let us consider a holonomic dynamical system with n_{dof} degrees of freedom whose configuration is expressed in terms of independent generalized coordinates $q_i, i = 1, 2, \ldots, n_{\text{dof}}$. Each q_i may be considered to be a component of a generalized displacement vector $\mathbf{q}$ in a n_{dof}-dimensional configuration space. Furthermore, we assume that all the generalized forces Q_i are associated with a conservative force field $\mathbf{Q} = -\partial_{\mathbf{q}} V$, where the potential energy $V(\mathbf{q}, t)$ is a function of $\mathbf{q}$ and time t. Let $T(\mathbf{q}, \dot{\mathbf{q}}, t)$, with $\dot{\mathbf{q}} = \mathrm{d}\mathbf{q}/\mathrm{d}t$, be the total kinetic energy of the system and $L = T - V$ the Lagrangian function. Then the standard form of Lagrange's equations may be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\partial_{\dot{\mathbf{q}}} L) - \partial_{\mathbf{q}} L = \mathbf{0}. \tag{1}$$

In general the application of Lagrange's equations yields a set of nonlinear differential equations of the form $\ddot{\mathbf{q}} + \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$, that is, n_{dof} *second-order* equations of motion. In view of our numerical developments we prefer the Hamiltonian formulation of the dynamical system in terms of $2n_{\text{dof}}$ *first-order* equations. The Hamiltonian function is defined by

$$H(\mathbf{q}, \mathbf{p}, t) = \mathbf{p} \cdot \dot{\mathbf{q}} - L(\mathbf{q}, \dot{\mathbf{q}}, t), \tag{2}$$

in which $\dot{\mathbf{q}}$ is implicitly expressed in terms of the generalized momentum vector $\mathbf{p}$ according to the relation $\mathbf{p} = \partial_{\dot{\mathbf{q}}} L$. The system of Lagrange's equations (1) is equivalent to the canonical equations of Hamilton given by

$$\begin{aligned} \dot{\mathbf{q}} &= \partial_{\mathbf{p}} H \\ \dot{\mathbf{p}} &= -\partial_{\mathbf{q}} H. \end{aligned} \tag{3}$$

Accordingly, the motion is described by means of $2n_{\text{dof}}$ *first-order* equations of motion expressed in terms of $2n_{\text{dof}}$ *independent* variables $(\mathbf{q}(t), \mathbf{p}(t))$, which are the coordinates of phase space.

In the present context we are especially interested in (scleronomic) natural systems where the kinetic energy is expressed as a homogeneous quadratic function of the p_is; that is, $T = \frac{1}{2}\mathbf{p} \cdot \mathbf{M}^{-1}\mathbf{p}$, where $\mathbf{M}(\mathbf{q})$ is the generalized inertia matrix. For natural systems the Hamiltonian function is equal to the total energy; that is,

$$H = T + V. \tag{4}$$

Hamilton's equations (3) imply that $\dot{H} = \partial_{\mathrm{t}} H$ such that for an autonomous natural system, where the Hamiltonian function does not depend explicitly on time ($\partial_{\mathrm{t}} H = 0$), the total energy is a constant of the motion.

3. PETROV–GALERKIN FINITE ELEMENT FORMULATION

In view of a Galerkin approximation of Hamilton's canonical equations (3) we form the weighted-residual statement for each equation in (3). Letting $\delta\mathbf{p}$ and $\delta\mathbf{q}$ be test functions sufficiently smooth on the time interval of interest $I = [t_0, t_0 + T]$, we have

$$\int_{t_0}^{t_0+T} [\{\dot{\mathbf{q}} - \partial_{\mathbf{p}} H\} \cdot \delta\mathbf{p} - \{\dot{\mathbf{p}} + \partial_{\mathbf{q}} H\} \cdot \delta\mathbf{q}]\, \mathrm{d}t = 0. \tag{5}$$

Note that (5) leads to the $2n_{\mathrm{dof}}$ canonical equations since both $\delta\mathbf{p}$ and $\delta\mathbf{q}$ are arbitrary. Equation (5) lies at the heart of the finite element method developed in the sequel.

Remark 3.1. Equation (5) can be related to Hamilton's law of varying action. To this end, integrate the term $\delta\mathbf{q} \cdot \dot{\mathbf{p}}$ by parts, which yields

$$\delta \int_{t_0}^{t_0+T} [\mathbf{p} \cdot \dot{\mathbf{q}} - H(\mathbf{p}, \mathbf{q}, t)]\, \mathrm{d}t - [\mathbf{p} \cdot \delta\mathbf{q}]_{t_0}^{t_0+T} = 0. \tag{6}$$

Here, the operator δ is to be interpreted as contemporaneous variation, such that $\delta H = \partial_{\mathbf{p}} H \cdot \delta\mathbf{p} + \partial_{\mathbf{q}} H \cdot \delta\mathbf{q}$. Alternatively, in the Lagrangian formulation, (6) can be written as

$$\delta \int_{t_0}^{t_0+T} L(\mathbf{q}, \dot{\mathbf{q}}, t)\, \mathrm{d}t - [\partial_{\dot{\mathbf{q}}} L \cdot \delta\mathbf{q}]_{t_0}^{t_0+T} = 0. \tag{7}$$

This equation is often called Hamilton's law of varying action (see, e.g., Williams [24], Appendix E). If one imposes the stationarity condition of vanishing $\delta\mathbf{q}$ at the endpoints t_0 and $t_0 + T$, (6) coincides with the modified Hamilton's principle (see, e.g., Goldstein [9], Chapter 8-5), and (7) coincides with Hamilton's principle.

3.1. Outline of the Time-Stepping Schemes

Consider a partition of the time interval of interest $I = [t_0, t_0 + T]$ into a number of finite elements of (time step) size $h_n = t_n - t_{n-1}$, such that $t_0 < t_1 < t_2 < \cdots < t_N = t_0 + T$ and

$$\int_{t_0}^{t_0+T} [\cdots]\, \mathrm{d}t = \sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} [\cdots]\, \mathrm{d}t. \tag{8}$$

On each subinterval $I_n = [t_{n-1},\ t_n]$, we consider piecewise smooth polynomial approximations of the trial functions $\mathbf{q}(t)$ and $\mathbf{p}(t)$, continuous across the element boundaries. In particular, we concentrate on the Lagrange family of finite elements consisting of polynomials of degree k. In addition to that, we employ piecewise smooth polynomial approximations of degree $k-1$ for the test functions $\delta\mathbf{q}(t)$ and $\delta\mathbf{p}(t)$, discontinuous across the element boundaries.

In the following we consider a representative finite element on I_n with $k+1$ nodes. For convenience of subsequent element calculations, we introduce a transformation to a master element using a local coordinate α, with its origin at the left endpoint and $\alpha = 1$ at

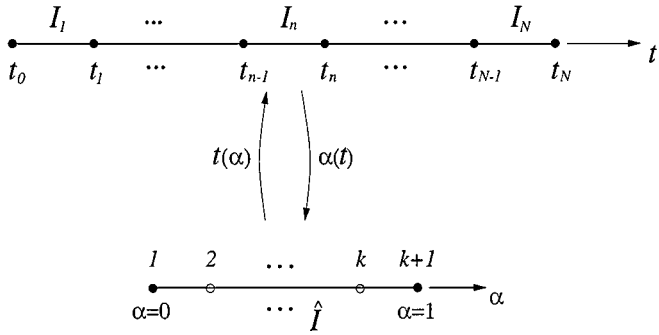

FIG. 1. Partition of the time interval $I = [t_0, t_0 + T]$ into N finite elements on the subintervals $I_n = [t_{n-1}, t_n]$ and master element I with $k + 1$ nodes.

the right endpoint, as shown in Fig. 1. Let $t \in I_n = [t_{n-1},\ t_n]$ transform to $\alpha \in \hat{I} = [0, 1]$, according to

$$\alpha(t) = \frac{t - t_{n-1}}{t_n - t_{n-1}}; \tag{9}$$

see also Fig. 1. Thus, for a $(k + 1)$-node element the domains of the global and local descriptions are related by the transformation α: $[t_{n-1},\ t_n] \to [\alpha_1, \alpha_{k+1}]$, such that $\alpha(t_{n-1}) = \alpha_1$ and $\alpha(t_n) = \alpha_{k+1}$.

The trial functions may now be approximated by continuous piecewise polynomials of degree k, according to

$$\mathbf{q}^h(\alpha) = \sum_{I=1}^{k+1} M_I(\alpha)\mathbf{q}_I$$

and

$$\mathbf{p}^h(\alpha) = \sum_{I=1}^{k+1} M_I(\alpha)\mathbf{p}_I, \tag{10}$$

where $M_I(\alpha)$ are nodal shape functions corresponding to Lagrange polynomials of degree k on the master element $\hat{I}$, given by

$$M_I(\alpha) = \prod_{\substack{J=1 \\ J\neq I}}^{k+1} \frac{\alpha - \alpha_J}{\alpha_I - \alpha_J}, \qquad 1 \le I \le k + 1. \tag{11}$$

Since $M_I(\alpha_K) = \delta_{IK}$, the Kronecker delta, the coefficients in (10) are the nodal values $\mathbf{q}_I = \mathbf{q}(\alpha_I)$ and $\mathbf{p}_I = \mathbf{p}(\alpha_I)$ of the generalized displacements and momenta, respectively. Note that the resulting global approximation of the trial functions remains C0 continuous.

Similarly, the test functions $\delta\mathbf{q}(\alpha)$ and $\delta\mathbf{p}(\alpha)$ are approximated by piecewise polynomials of reduced degree $k - 1$, according to

$$\delta\mathbf{q}^h(\alpha) = \sum_{I=1}^{k} \tilde{M}_I(\alpha)\delta\mathbf{q}_I$$

and

$$\delta \mathbf{p}^h(\alpha) = \sum_{I=1}^{k} \tilde{M}_I(\alpha)\delta \mathbf{p}_I, \tag{12}$$

where $\tilde{M}_I(\alpha)$ are reduced shape functions consisting of polynomials of degree $k-1$. Note that the finite element approximation of the test functions leads to possible discontinuities across the element boundaries; i.e., at time t_n, there may exist jumps $[\![\delta \mathbf{q}_n^h]\!]$ and $[\![\delta \mathbf{p}_n^h]\!]$, where $[\![\{\bullet\}_n]\!] = \lim_{\varepsilon \to 0^+}[\{\bullet\}(t_n+\varepsilon) - \{\bullet\}(t_n-\varepsilon)]$. From now on we omit the superscript h without danger of confusion. We may write

$$\mathbf{q}'(\alpha) = \sum_{I=1}^{k+1} M_I'(\alpha)\mathbf{q}_I = \sum_{I=1}^{k} \tilde{M}_I(\alpha)\tilde{\mathbf{q}}_I, \tag{13}$$

where $\{\bullet\}' = \mathrm{d}\{\bullet\}/\mathrm{d}\alpha$ and $\tilde{\mathbf{q}}_I$ are linear combinations of the $\mathbf{q}_I$s. We refer to Table I for examples involving nodal shape functions $M_I(\alpha)$ of polynomial degree $1 \le k \le 3$, along with corresponding shape functions $\tilde{M}_I(\alpha)$ and the related $\tilde{\mathbf{q}}_I$s as combinations of the nodal values $\mathbf{q}_I$. Analogous to (13), we may write

$$\mathbf{p}'(\alpha) = \sum_{I=1}^{k+1} M_I'(\alpha)\mathbf{p}_I = \sum_{I=1}^{k} \tilde{M}_I(\alpha)\tilde{\mathbf{p}}_I, \tag{14}$$

where again the $\tilde{\mathbf{p}}_I$s are linear combinations of the nodal values $\mathbf{p}_I$, analogous to the relations between the $\tilde{\mathbf{q}}_I$s and the $\mathbf{q}_I$s given in Table I.

Due to the fact that the resulting global approximation of the test functions allows interelement discontinuities we obtain a recursive time-stepping scheme. Since the global approximation of the trial functions is continuous, the formulation belongs to a Petrov–Galerkin method where the trial and test spaces are different.

Next we introduce the finite element approximations (10) and (12) into the weak form (5) of Hamilton's equations. With regard to the arbitrariness of the $\delta \mathbf{q}_I$s and $\delta \mathbf{p}_I$s on each

TABLE I

Nodal Shape Functions $M_I(\alpha)$ for Polynomial Approximations of Degrees $k = 1$, $k = 2$, and $k = 3$ along with Shape Functions $\tilde{M}_I(\alpha)$ and Associated Values $\mathbf{q}_I$

	$M_I(\alpha)$	$\tilde{M}_I(\alpha)$	$\tilde{\mathbf{q}}_I$ ($\tilde{\mathbf{p}}_I$ analogously)
$k=1$	$M_1 = 1-\alpha$ $M_2 = \alpha$	$\tilde{M}_1 = 1$	$\tilde{\mathbf{q}} = \mathbf{q}_2 - \mathbf{q}_1$
$k=2$	$M_I = [2\alpha-1][\alpha-1]$ $M_2 = -4[\alpha^2-\alpha]$ $M_3 = [2\alpha-1]\alpha$	$\tilde{M}_1 = 1-\alpha$ $\tilde{M}_2 = \alpha$	$\tilde{\mathbf{q}}_1 = -3\mathbf{q}_1 + 4\mathbf{q}_2 - \mathbf{q}_3$ $\tilde{\mathbf{q}}_2 = \mathbf{q}_1 - 4\mathbf{q}_2 + 3\mathbf{q}_3$
$k=3$	$M_1 = -\frac{9}{2}[\alpha-\frac{1}{3}][\alpha-\frac{2}{3}][\alpha-1]$ $M_2 = \frac{27}{2}[\alpha-\frac{2}{3}][\alpha-1]\alpha$ $M_3 = -\frac{27}{2}[\alpha-\frac{1}{3}][\alpha-1]\alpha$ $M_4 = \frac{9}{2}[\alpha-\frac{1}{3}][\alpha-\frac{2}{3}]\alpha$	$\tilde{M}_1 = [2\alpha-1][\alpha-1]$ $\tilde{M}_2 = -4[\alpha^2-\alpha]$ $\tilde{M}_3 = [2\alpha-1]\alpha$	$\tilde{\mathbf{q}}_1 = -\frac{11}{2}\mathbf{q}_1 + 9\mathbf{q}_2 - \frac{9}{2}\mathbf{q}_3 + \mathbf{q}_4$ $\tilde{\mathbf{q}}_2 = \frac{1}{8}\mathbf{q}_1 - \frac{27}{8}\mathbf{q}_2 + \frac{27}{8}\mathbf{q}_3 - \frac{1}{8}\mathbf{q}_4$ $\tilde{\mathbf{q}}_3 = -\mathbf{q}_1 + \frac{9}{2}\mathbf{q}_2 - 9\mathbf{q}_3 + \frac{11}{2}\mathbf{q}_4$

subinterval I_n, we obtain the following system of equations,

$$\begin{aligned}\sum_{J=1}^{k}\int_0^1 \tilde{M}_I\tilde{M}_J\,\mathrm{d}\alpha\,\tilde{\mathbf{q}}_J - h_n\int_0^1 \tilde{M}_I\,\partial_{\mathbf{p}}H\,\mathrm{d}\alpha = \mathbf{0}\\ \sum_{J=1}^{k}\int_0^1 \tilde{M}_I\tilde{M}_J\,\mathrm{d}\alpha\,\tilde{\mathbf{p}}_J - h_n\int_0^1 \tilde{M}_I\,\partial_{\mathbf{q}}H\,\mathrm{d}\alpha = \mathbf{0},\end{aligned} \tag{15}$$

for $I = 1, \ldots, k$. Depending on the chosen polynomial degree k of the finite element approximation, the equations in (15) furnish distinct recurrence formulas for the calculation of the nodal variables $\mathbf{q}_I$ and $\mathbf{p}_I$, for $I = 2, \ldots, k+1$. Moreover, on each successive time interval $I_n = [t_{n-1}, t_n]$, the nodal quantities at time t_{n-1}, i.e., $\mathbf{q}_1 = \mathbf{q}(\alpha(t_{n-1}))$ and $\mathbf{p}_1 = \mathbf{p}(\alpha(t_{n-1}))$, are given due to the global continuity of the trial functions.

3.2. Algorithmic Conservation Properties

Next we show that the time finite element formulation developed above inherently conserves the Hamiltonian function in the case of autonomous systems. In addition to that we show that the resulting time-stepping schemes automatically preserve the generalized momenta corresponding to cyclic coordinates.

Conservation of the Hamiltonian. Scalar multiplication of $(15)_1$ and $(15)_2$ with $\tilde{\mathbf{p}}_I$ and $\tilde{\mathbf{q}}_I$, respectively, and subsequent summation yields

$$\begin{aligned}\sum_{I,J=1}^{k}\int_0^1 \tilde{M}_I\tilde{M}_J\,\mathrm{d}\alpha\,\tilde{\mathbf{p}}_I\cdot\tilde{\mathbf{q}}_J - h_n\sum_{I=1}^{k}\int_0^1 \tilde{M}_I\,\partial_{\mathbf{p}}H\,\mathrm{d}\alpha\cdot\tilde{\mathbf{p}}_I = 0\\ -\sum_{I,J=1}^{k}\int_0^1 \tilde{M}_I\tilde{M}_J\,\mathrm{d}\alpha\,\tilde{\mathbf{p}}_J\cdot\tilde{\mathbf{q}}_I - h_n\sum_{I=1}^{k}\int_0^1 \tilde{M}_I\,\partial_{\mathbf{q}}H\,\mathrm{d}\alpha\cdot\tilde{\mathbf{q}}_I = 0.\end{aligned} \tag{16}$$

Addition of $(16)_1$ and $(16)_2$ leads to

$$\sum_{I=1}^{k}\left[\int_0^1 \tilde{M}_I\,\partial_{\mathbf{q}}H\,\mathrm{d}\alpha\cdot\tilde{\mathbf{q}}_I + \int_0^1 \tilde{M}_I\,\partial_{\mathbf{p}}H\,\mathrm{d}\alpha\cdot\tilde{\mathbf{p}}_I\right] = 0. \tag{17}$$

Referring to (13) and (14), (17) may also be written in the form

$$\int_0^1 [\partial_{\mathbf{q}}H\cdot\mathbf{q}'(\alpha) + \partial_{\mathbf{p}}H\cdot\mathbf{p}'(\alpha)]\,\mathrm{d}\alpha = 0. \tag{18}$$

On the other hand, the Fundamental Theorem of Calculus implies

$$\begin{aligned}H_n - H_{n-1} = H(\mathbf{q}(\alpha),\mathbf{p}(\alpha),\alpha)\big|_{\alpha=0}^{\alpha=1} &= \int_0^1\left[\frac{\mathrm{d}}{\mathrm{d}\alpha}H(\mathbf{q}(\alpha),\mathbf{p}(\alpha),\alpha)\right]\mathrm{d}\alpha\\ &= \int_0^1 [\partial_{\mathbf{q}}H\cdot\mathbf{q}'(\alpha) + \partial_{\mathbf{p}}H\cdot\mathbf{p}'(\alpha) + \partial_\alpha H]\,\mathrm{d}\alpha.\end{aligned} \tag{19}$$

Thus (18) leads to

$$H_n = H_{n-1} + \int_{t_{n-1}}^{t_n} \partial_t H \, \mathrm{d}t. \tag{20}$$

Accordingly, in the autonomous case, where $H = H(\mathbf{q}, \mathbf{p})$ and therefore $\partial_t H = 0$, the algorithmic phase flow generated by (15) preserves the Hamiltonian function in the sense that $H_n = H_{n-1}$. That is, for any polynomial degree k of the finite element approximation, the Hamiltonian is conserved at the end of each successive time interval $I_n = [t_{n-1}, t_n]$.

Cyclic coordinates. To see what happens if cyclic coordinates appear, consider $(15)_2$, which, after summation over $I = 1, \ldots, k$, yields

$$\int_0^1 \sum_{I=1}^{k} \tilde{M}_I \sum_{J=1}^{k} \tilde{M}_J \tilde{\mathbf{p}}_J \, \mathrm{d}\alpha + h_n \int_0^1 \sum_{I=1}^{k} \tilde{M}_I \partial_{\mathbf{q}} H \, \mathrm{d}\alpha = \mathbf{0}. \tag{21}$$

Since the Lagrangean shape functions fulfill the relation $\sum_{I=1}^{k} \tilde{M}_I = 1$ and, in view of (14), $\mathbf{p}'(\alpha) = \sum_{I=1}^{k} \tilde{M}_I(\alpha) \tilde{\mathbf{p}}_I$, Eq. (21) leads to

$$\mathbf{p}(1) - \mathbf{p}(0) = -h_n \int_0^1 \partial_{\mathbf{q}} H \, \mathrm{d}\alpha. \tag{22}$$

Hence, if q_i is a cyclic coordinate, i.e., $\partial H / \partial q_i = 0$, then it follows from (22) that the generalized momentum p_i associated with q_i is conserved by the algorithm in the sense that $p_i(\alpha(t_n)) = p_i(\alpha(t_{n-1}))$.

4. ONE-DIMENSIONAL MODEL PROBLEM

In this section we apply the general method developed above to the case of one-dimensional motion. In particular, we give a detailed account of the numerical implementation of the time-stepping algorithms emanating from the formulas in (15) for polynomial degrees $k = 1$, $k = 2$, and $k = 3$. We focus our numerical experiments on the accuracy and energy conserving property of the time-stepping schemes. In this context the specific quadrature rules employed play an important role.

Circular pendulum. We consider the motion of a particle of mass m suspended by a massless rod of length l (see Fig. 2). In particular, we investigate the oscillatory motion with initial conditions $q = \pi/2$ and $p = 0$. The kinetic energy of the particle is $T = \frac{1}{2} m l^2 \dot{q}^2$ and the potential energy may be written as $V = -mgl \cos q$, such that the Lagrangian function is $L(q, \dot{q}) = \frac{1}{2} m l^2 \dot{q}^2 + mgl \cos q$. Lagrange's equation furnishes the equation of motion in the form $\ddot{q} + g/l \sin q = 0$. Furthermore, we obtain $p = \partial_{\dot{q}} L = m l^2 \dot{q}$, such that the Hamiltonian function, being equal to the total energy, takes the form

$$H(q, p) = T(p) + V(q) = \frac{p^2}{2ml^2} - mgl \cos q. \tag{23}$$

Now the expressions $\partial_q H = \partial_q V$ and $\partial_p H = p/[ml^2]$, can be inserted into the formulas of the temporal finite element method in (15). Referring to (10), we have $p(\alpha) = \sum_{I=1}^{k+1} M_I(\alpha)$

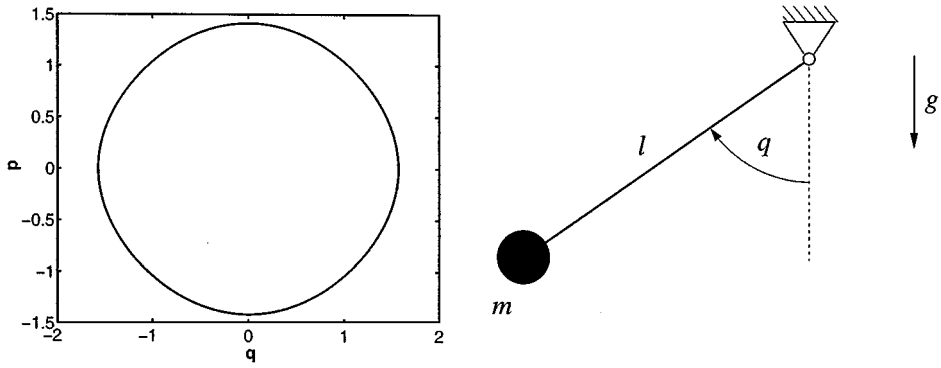

FIG. 2. Circular pendulum and phase curve corresponding to initial conditions $q = \pi/2$ and $p = 0$.

p_I, such that (15) may be written in the form

$$\begin{aligned} \sum_{J=1}^{k} \int_0^1 \tilde{M}_I \tilde{M}_J \, \mathrm{d}\alpha \, \tilde{q}_J - \frac{h_n}{ml^2} \sum_{J=1}^{k+1} \int_0^1 \tilde{M}_I \tilde{M}_J \, \mathrm{d}\alpha \, p_J = 0 \\ \sum_{J=1}^{k} \int_0^1 \tilde{M}_I \tilde{M}_J \, \mathrm{d}\alpha \, \tilde{p}_J + h_n \int_0^1 \tilde{M}_I \partial_q V \, \mathrm{d}\alpha = 0 \end{aligned} \tag{24}$$

for $I = 1, \ldots, k$ and with $\partial_q V = mgl \sin q$. The equations in (24) constitute the foundation of a family of implicit one-step methods. Essentially two additional steps have to be performed in order to obtain a particular time-stepping scheme.

1. The selection of the polynomial degree k leads to the corresponding finite element formulation. Concerning the integrals in (24) involving only the shape functions M_J and $\tilde{M}_I$, exact integration can be readily performed.

2. Eventually, a specific time-stepping algorithm is completely defined by the evaluation of $\int_0^1 \tilde{M}_I \partial_q V \, \mathrm{d}\alpha$ in (24). In what follows we shall investigate the influence of *numerical quadrature* on the accuracy and energy conserving property of the related algorithm. This point is of crucial interest especially with regard to dynamical systems with more than one degree of freedom where exact integration is rarely feasible.

4.1. Linear Elements

First we consider the case $k = 1$ corresponding to linear trial functions. Referring to Table I and (24), we obtain the particular finite element formulation given by

$$\begin{aligned} q_2 - q_1 - \frac{h_n}{2ml^2}[p_1 + p_2] = 0 \\ P_2 - p_1 + h_n \int_0^1 \partial_q V \, d\alpha = 0, \end{aligned} \tag{25}$$

with $\partial_q V = mgl \sin q$. Recall that, according to the notation introduced above, the nodal quantities (q_1, p_1) are associated with time t_{n-1} and (q_2, p_2) are associated with time t_n;

furthermore, $h_n = t_n - t_{n-1}$. The time-stepping algorithm is completely specified by the quadrature employed for the evaluation of the integral in (25).

Exact quadrature. The exact calculation of the integral in (25) does not pose any difficulties in the present one-dimensional context. In fact, the following identity holds,

$$\int_0^1 \partial_q V \, \mathrm{d}\alpha = \frac{V_2 - V_1}{q_2 - q_1}, \tag{26}$$

where $V_1 = V(q_1)$, the potential energy at time t_{n-1}. Analogously, $V_2 = V(q_2)$, the potential energy at time t_n. The validity of (26) follows from the Fundamental Theorem of Calculus, which implies

$$\begin{aligned} V_2 - V_1 = V(q(\alpha))\big|_{\alpha=0}^{\alpha=1} &= \int_0^1 \left[\frac{\mathrm{d}}{\mathrm{d}\alpha} V(q(\alpha))\right] \mathrm{d}\alpha \\ &= \int_0^1 \partial_q V q'(\alpha) \, \mathrm{d}\alpha = \int_0^1 \partial_q V \, \mathrm{d}\alpha [q_2 - q_1], \end{aligned} \tag{27}$$

where the relation $q'(\alpha) = q_2 - q_1$ has been used, which holds in the present case $k = 1$. Accordingly, the time-stepping scheme in (25) may now be written alternatively as

$$\begin{aligned} q_2 - q_1 - \frac{h_n}{2ml^2}[p_1 + p_2] &= 0 \\ p_2 - p_1 + h_n \frac{V_2 - V_1}{q_2 - q_1} &= 0, \end{aligned} \tag{28}$$

which reveals a surprising result: The time-stepping scheme in (28) coincides with the method of Greenspan [12] (cf. Eqs. (3.5) and (3.6) in [12]). Greenspan's method covers conservative one-dimensional initial-value problems of the form $\ddot{q} = f(q)$, with $f = -\mathrm{d}V/\mathrm{d}q$, and relies on a difference method which, by design, is energy conserving.

In the present case the energy conserving property of the algorithm (28) follows directly from (20).

Numerical quadrature. In analogy to the spatial finite element method we next investigate the approximation of the integral in (25) by means of different quadrature rules. In this context the main question is how the accuracy and the conservation properties of the respective time-stepping method will be affected by the quadrature.

Midpoint rule. Let us consider the midpoint approximation of the integral in (25) given by

$$\int_0^1 \partial_q V \, \mathrm{d}\alpha \approx \partial_q V(q(1/2)), \tag{29}$$

where $q(1/2)$ is the midpoint value of q, i.e., at $\alpha = 1/2$, such that $q(1/2) = \frac{1}{2}[q_1 + q_2]$. The resulting time-stepping scheme reads

$$\begin{aligned} q_2 - q_1 - \frac{h_n}{2ml^2}[p_1 + p_2] &= 0 \\ p_2 - p_1 + h_n \partial_q V(q(1/2)) &= 0. \end{aligned} \tag{30}$$

Accordingly, the algorithmic form (30) coincides with the midpoint rule applied to the considered nonlinear problem; i.e., $\ddot{q} + g/l \sin q = 0$. It is well-known that the midpoint rule is not energy conserving in the nonlinear regime (see, e.g., Simo *et al.* [21]). We refer to Section 4.1.3 for related numerical investigations.

Trapezoidal rule. Alternatively, the computation of the integral in (25) by means of the trapezoidal rule yields

$$\int_0^1 \partial_q V \, \mathrm{d}\alpha \approx \frac{1}{2}[\partial_q V(q_1) + \partial_q V(q_2)], \tag{31}$$

such that the corresponding time-stepping algorithm is given by

$$\begin{aligned} q_2 - q_1 - \frac{h_n}{2ml^2}[p_1 + p_2] &= 0 \\ p_2 - p_1 + h_n \frac{1}{2}[\partial_q V(q_1) + \partial_q V(q_2)] &= 0. \end{aligned} \tag{32}$$

It can be easily verified that the time-stepping scheme (32) coincides with the average acceleration method applied to the considered nonlinear problem; i.e., $\ddot{q} + g/l \sin q = 0$. It is well-known that the average acceleration method is not energy conserving in the nonlinear regime (see, e.g., Hughes [14]). We refer to Section 4.1.3 for related numerical investigations.

Gaussian quadrature. In analogy to customary spatial finite element formulations, the numerical evaluation of the integral in (25) may be accomplished by choosing Gaussian quadrature rules, such that

$$\int_0^1 \partial_q V \, \mathrm{d}\alpha \approx \sum_{l=1}^{N_l} \partial_q V(q(\xi_l)) w_l, \tag{33}$$

where w_l and ξ_l are the weights and abscissae for $[0, 1]$. Recall that the Gauss rule of order N_l integrates exactly polynomials of degree $2N_l - 1$. With increasing order of the employed Gauss rule we expect an associated progressively energy conserving time-stepping scheme. This expectation is confirmed by our numerical results documented in Section 4.1.3.

4.1.1. *Numerical Implementation*

Let us consider a typical subinterval $I_n = [t_{n-1}, t_n]$ with the corresponding master element on $\hat{I} = [0, 1]$. In the case of $k = 1$ there are two nodes located at $\alpha = 0$ and $\alpha = 1$. The nodal values of the generalized displacements and momenta at $\alpha = 0$, that is q_1 and p_1, are given quantities at time t_{n-1}. The nodal unknowns at time t_n, that is q_2 and p_2 in the local description, may now be calculated by employing the time-stepping algorithm emanating from (25). Substitution from $(25)_1$ for p_2 into $(25)_2$ leads to the residual

$$R(q_2) = \frac{2ml^2}{h_n}[q_2 - q_1] - 2p_1 + h_n \int_0^1 \partial_q V \, \mathrm{d}\alpha = 0, \tag{34}$$

which is a nonlinear function of q_2. The iterative solution by means of Newton's method is summarized in Table II. The evaluation of the integrals in Table II may be accomplished

TABLE II

One-Dimensional Motion: Summary of Computations for One Typical Time Step in the Case $k = 1$ Corresponding to Linear Finite Elements in Time

Given: initial conditions: $q_1,\ p_1$; time step size: h_n; set iteration counter: $i = 1$

Find: nodal unknows q_2 and p_2

(a) Compute residual

$$R\left(q_2^{(i)}\right) = \frac{2ml^2}{h_n}\left[q_2^{(i)} - q_1\right] - 2p_1 + h_n \int_0^1 \partial_q V^{(i)}\, \mathrm{d}\alpha$$

if $\left|R\left(q_2^{(i)}\right)\right| > \varepsilon$ goto (b) else goto (c)

(b) Compute tangent

$$K\left(q_2^{(i)}\right) = \frac{2ml^2}{h_n} + h_n \int_0^1 M_2 \partial^2_{qq} V^{(i)}\, \mathrm{d}\alpha$$

Solve for increment Δq_2

$$\Delta q_2 = -K^{-1}\left(q_2^{(i)}\right) R\left(q_2^{(i)}\right)$$

Update generalized displacement

$$q_2^{(i+1)} = q_2^{(i)} + \Delta q_2$$

goto (a) with $i = i + 1$

(c) Update generalized momentum

$$p_2^{(i)} = \frac{2ml^2}{h_n}\left[q_2^{(i)} - q_1\right] - p_1$$

Note. Circular pendulum: $V = -mgl\cos q$.

by using one of the quadratures discussed above. Accordingly, depending on the particular quadrature rule, Table II comprises a family of time-stepping schemes associated with the linear finite element formulation ($k = 1$).

Remark 4.1. In Table II, $\varepsilon \to 0$ is the numerical tolerance applied in the iterative solution procedure. In the case of exact quadrature, that is when formula (26) is used, the following relation can be easily verified:

$$H_n - H_{n-1} = \frac{1}{h_n} R(q_2)[q_2 - q_1].$$

Accordingly, algorithmic energy conservation is automatically attained when the iterative solution procedure has converged. This statement is a direct consequence of the general result (20).

4.1.2. *Numerical Accuracy and Quadrature*

Next we investigate the accuracy of the time-stepping schemes related to the linear finite element formulation ($k = 1$). To this end we consider the error in the generalized displacements and momenta, respectively, defined as the difference between the reference and the approximate solutions,

$$e_q(t) = q_*(t) - q_h(t)$$

and

$$e_p(t) = p_*(t) - p_h(t). \tag{35}$$

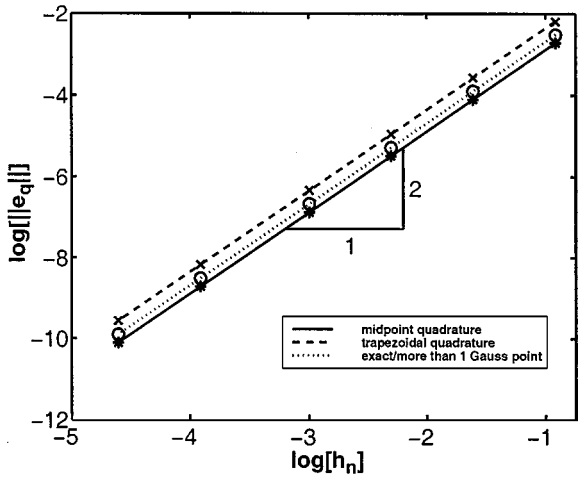

FIG. 3. Linear elements ($k = 1$): Computed mean-square norm of the error in the generalized displacements for different time-stepping schemes associated with specific quadrature rules.

As reference solutions $q_*(t)$ and $p_*(t)$ we consider the numerical results obtained with very small, equally spaced, time steps of size $h_n = 0.0001$. Within the time interval $I = [0, T]$, with $T = 8$, we calculate the mean-square norm of the error according to

$$\|e\|_{L_2(0,T)} = \left\{ \int_0^T e^2 \, \mathrm{d}t \right\}^{1/2} . \tag{36}$$

Using *exact quadrature* in Table II by employing formula (26) we obtain the numerical results for the error in the generalized displacements depicted in Fig. 3. Additionally, the results obtained by applying the *midpoint quadrature*, the *trapezoidal rule*, as well as the *two-point Gaussian quadrature* are also shown in Fig. 3. Accordingly, the rate of convergence of the investigated time-stepping schemes is two with respect to the mean-square norm. The specific quadrature rule used has only minor influence on the numerical accuracy of the related time-stepping scheme. Referring to Fig. 4, the same conclusions can be drawn concerning the error in the generalized momenta.

4.1.3. *Energy Conservation and Quadrature*

As has been shown in Section 3.2, exact integration in Table II implies exact algorithmic conservation of the total energy. In this section we investigate how different quadrature rules effect the energy conservation property. Specifically we consider the midpoint, trapezoidal, and Gaussian quadrature rules. Naturally, we expect progressively energy conserving algorithms with increasing accuracy of the used quadrature rules. We measure the error in the total energy according to

$$e_H = \sum_{n=1}^{N} |H_* - H_n| h_n , \tag{37}$$

where $h_n = t_n - t_{n-1}$ is the time step size, H_n is the calculated total energy at time t_n, and H_* is the exact value of the constant total energy. As before the time interval of interest is $I = [0, T]$, with $T = 8$, and N denotes the number of (equally spaced) time steps.

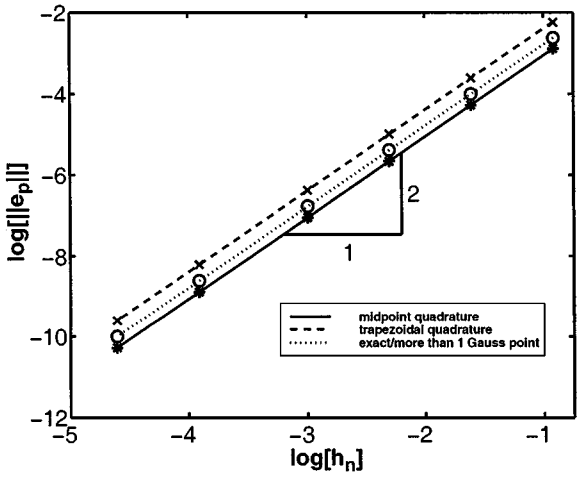

FIG. 4. Linear elements ($k = 1$): Computed mean-square norm of the error in the generalized momenta for different time-stepping schemes associated with specific quadrature rules.

Exact quadrature. Our numerical experiments confirm the exact energy preserving property of the developed algorithm, independent of the time step size, when exact integration is used by employing formula (26). Our computations rely on a floating point relative accuracy of $2.22e-16$. Taking into account numerical round-off in the calculation of (37), we consider the value $\log[e_H] \approx -24$ as the numerical limit corresponding to $e_H = 0$. Accordingly, the results obtained by using exact quadrature are represented by the horizontal straight line in Fig. 5.

Midpoint and trapezoidal rules. Figure 5 shows that application of the midpoint rule (see formula (29)) or the trapezoidal rule (see formula (31)) leads to time-stepping schemes violating energy preservation of the underlying conservative dynamical system. Both quadrature rules yield a straight line of slope 2 in the log-log plot.

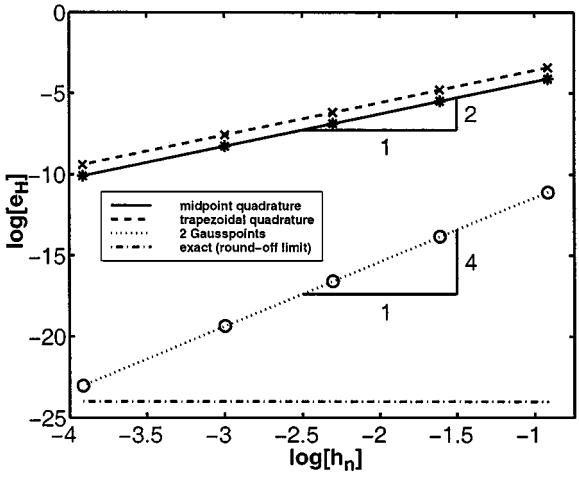

FIG. 5. Linear elements ($k = 1$): Computed error in the total energy for different time-stepping schemes associated with specific quadrature rules.

Gaussian quadrature. Using two Gauss points in (33) we obtained a straight line of slope 4 as shown in the log-log plot in Fig. 5. Already three Gauss points essentially yield results being indistinguishable from the exact integration. In summary, with increasing order of the quadrature rule the corresponding time-stepping scheme progressively inherits the energy conserving property of the underlying conservative system.

4.2. Quadratic Elements

In this section we investigate the case $k=2$ corresponding to quadratic trial functions. Referring to Table I and (24), we obtain the particular finite element formulation given by

$$\begin{aligned}
-5q_1+4q_2+q_3-\frac{h_n}{ml^2}[p_1+2p_2]&=0\\
-q_1-4q_2+5q_3-\frac{h_n}{ml^2}[2p_2+p_3]&=0\\
\frac{5}{6}p_1-\frac{2}{3}p_2-\frac{1}{6}p_3-h_n\int_0^1\tilde{M}_1\partial_q V\,\mathrm{d}\alpha&=0\\
\frac{1}{6}p_1+\frac{2}{3}p_2-\frac{5}{6}p_3-h_n\int_0^1\tilde{M}_2\partial_q V\,\mathrm{d}\alpha&=0,
\end{aligned}\tag{38}$$

with $\partial_q V=mgl\sin q$. Again $h_n=t_n-t_{n-1}$, and the generalized displacement q_1 as well as the generalized momentum p_1 are given quantities at time t_{n-1}. The nodal unknowns q_2, q_3, and p_2, p_3 corresponding to the generalized displacements and momenta at time $t_{n-\frac{1}{2}}$ and t_n can be calculated by using the finite element formulation in (38). As in the case $k=1$ treated above, the additional specification of the quadrature employed for the evaluation of the integrals in (38) completely defines a particular time-stepping scheme. Here we concentrate on *Gaussian quadrature* for the computation of the integrals in (38). In this connection it is interesting to note that using two Gauss points leads to a scheme which coincides with the two-stage Gauss Runge–Kutta method (see Hulme [16] and the Appendix).

4.2.1. *Numerical Implementation*

Using $(38)_1$ and $(38)_2$, p_2 and p_3 can be eliminated from $(38)_3$ and $(38)_4$, such that we obtain the residual vector

$$\mathbf{R}(q_2,q_3)=\begin{bmatrix}\frac{ml^2}{h_n}[q_3-q_1]-p_1+h_n\int_0^1\tilde{M}_1\partial_q V\,\mathrm{d}\alpha\\ \frac{ml^2}{h_n}[3q_3-8q_2+5q_1]+p_1+h_n\int_0^1\tilde{M}_2\partial_q V\,\mathrm{d}\alpha\end{bmatrix},\tag{39}$$

which is a nonlinear function of q_2 and q_3. An outline of the iterative solution procedure based on Newton's method is given in Table III. The integrals in Table III may be evaluated by employing Gaussian quadrature, analogous to (33).

Remark 4.2. Analogous to Remark 4.1 the energy conserving property of the algorithm in Table III associated with exact quadrature can be verified by a straightforward calculation. Scalar multiplication of the residual vector $\mathbf{R}=\mathbf{R}(q_2,q_3)=[R_1,R_2]^T$ with the

TABLE III

One-Dimensional Motion: Summary of Computations for One Typical Time Step in the Case $k = 2$ Corresponding to Quadratic Finite Elements in Time

Given: initial conditions: $q_1,\ p_1$; time step size: h_n; set iteration counter: $i = 1$

$$\mathbf{R}_1 = \frac{ml^2}{h_n}\begin{bmatrix} -q_1 \\ 5q_1 \end{bmatrix} + \begin{bmatrix} -p_1 \\ p_1 \end{bmatrix}, \quad \mathbf{K}_1 = \frac{ml^2}{h_n}\begin{bmatrix} 0 & 1 \\ -8 & 3 \end{bmatrix}$$

Find: vector of nodal unknowns

$$\hat{\mathbf{q}} = [q_2, q_3]^T \text{ and } \hat{\mathbf{p}} = [p_2,\ p_3]^T$$

(a) Compute residual

$$\mathbf{R}^{(i)} = \mathbf{R}_1 + \mathbf{K}_1\hat{\mathbf{q}}^{(i)} + \mathbf{R}_2^{(i)}$$

where

$$\mathbf{R}_2^{(i)} = \left[R_1^{(i)},\ R_2^{(i)}\right]^T \text{ with } R_J^{(i)} = h_n \int_0^1 \tilde{M}_I\, \partial_q V^{(i)}\, \mathrm{d}\alpha$$

if $|\mathbf{R}^{(i)}| > \varepsilon$ goto (b) else goto (c)

(b) Compute tangent

$$\mathbf{K}^{(i)} = \mathbf{K}_1 + \mathbf{K}_2^{(i)}$$

where

$$\mathbf{K}_2^{(i)} = \begin{bmatrix} k_{11}^{(i)} & k_{12}^{(i)} \\ k_{21}^{(i)} & k_{22}^{(i)} \end{bmatrix} \text{ with } k_{IJ}^{(i)} = h_n \int_0^1 \tilde{M}_I M_{J+1} \partial_{qq}^2 V^{(i)}\, \mathrm{d}\alpha$$

Solve for increments

$$\Delta\hat{\mathbf{q}} = -{\mathbf{K}^{(i)}}^{-1}\mathbf{R}^{(i)}$$

Update generalized displacements

$$\hat{\mathbf{q}}^{(i+1)} = \hat{\mathbf{q}}^{(i)} + \Delta\hat{\mathbf{q}}$$

goto (a) with $i = i + 1$

(c) Update generalized momenta

$$p_2^{(i)} = \frac{ml^2}{2h_n}\left[q_3^{(i)} + 4q_2^{(i)} - 5q_1\right] - \frac{1}{2}p_1$$

$$p_3^{(i)} = \frac{ml^2}{h_n}\left[4q_3^{(i)} - 8q_2^{(i)} + 4q_1\right] + p_1$$

Note. Circular pendulum: $V = -mgl\cos q$.

vector $[\tilde{q}_1, \tilde{q}_2]^T$ yields

$$H_n - H_{n-1} = \frac{1}{h_n}\mathbf{R}(q_2, q_3) \cdot \begin{bmatrix} -3q_1 + 4q_2 - q_3 \\ q_1 - 4q_2 + 3q_3 \end{bmatrix},$$

which can be regarded as the computational counterpart of the general expression (20) for the algorithmic conservation of the Hamiltonian.

4.2.2. *Numerical Accuracy and Quadrature*

In this section we examine the numerical accuracy of the time-stepping algorithms corresponding to quadratic finite elements in time ($k = 2$). In particular, we elaborate on the influence of the employed (Gaussian) quadrature rule on the numerical accuracy of the associated time-stepping scheme. As in Section 4.1.2, we calculate the mean-square norm

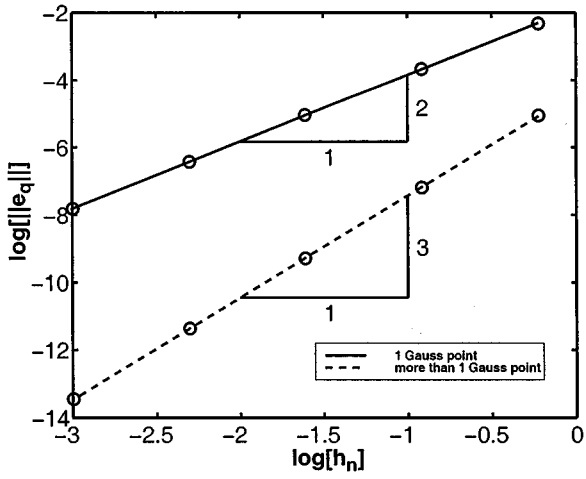

FIG. 6. Quadratic elements ($k = 1$): Computed mean-square norm of the error in the generalized displacements for specific time-stepping schemes associated with different Gaussian quadrature rules.

of the error in the generalized displacements and momenta (see Eq. (36)). Again the time interval of interest is $I = [0, T]$, with $T = 8$. The reference solution is obtained with linear elements incorporating exact quadrature and a time step size of $h_n = 0.0001$.

The calculated results for the mean-square norm of the error in the generalized displacements are depicted in Fig. 6. Accordingly, the application of only one Gauss point for the evaluation of the integrals in Table III yields an associated time-stepping scheme with second-order accuracy. Already two Gauss points lead to a third-order accurate time-stepping scheme. Our numerical calculations show that the application of more than two Gauss points does not make any difference from the accuracy point of view.

4.2.3. *Energy Conservation and Quadrature*

In analogy to the case $k = 1$ in Section 4.1.3 we examine how the employed Gaussian quadrature influences the algorithmic energy conservation for $k = 2$. According to (20) exact quadrature implies exact energy conservation of the associated time-stepping scheme. Accordingly, with increasing order of the employed Gauss rule we expect a progressively energy conserving time-stepping algorithm. This feature is verified in view of Fig. 7, where again the error in the energy according to (37) is shown. Accordingly, already four Gauss points yield results being almost indistinguishable from exact quadrature.

4.3. Cubic Elements

In this section we investigate the case $k = 3$ corresponding to cubic trial functions. Referring to (24) along with the corresponding values for $M_I(\alpha)$, $\tilde{M}_I(\alpha)$, $\tilde{q}_I$, and $\tilde{p}_I$, summarized in Table I, one obtains the finite element formulation for $k = 3$. Again the selection of a particular quadrature rule for the evaluation of the integrals containing the potential energy function completely defines a specific time-stepping scheme. In the present case we will concentrate on Gaussian quadrature. Interestingly, if three Gauss points are employed one obtaines a scheme that coincides with the three-stage Gauss Runge–Kutta method (see Hulme [16] and the Appendix).

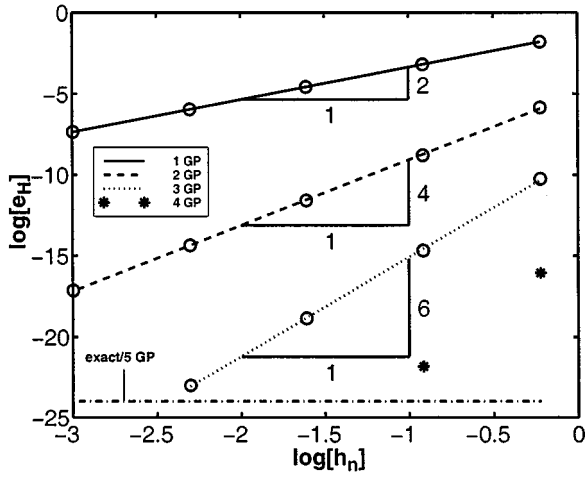

FIG. 7. Quadratic elements ($k=2$): Computed error in the total energy for different time-stepping schemes associated with specific quadrature rules.

4.3.1. *Numerical Implementation*

Within each consecutive subinterval $I_n=[t_{n-1}, t_n]$ one has to solve for the nodal unknowns $\hat{\mathbf{q}}=[q_2, q_3, q_4]$ and $\hat{\mathbf{p}}=[p_2, p_3, p_4]$, while q_1 and p_1 are given quantities at time t_{n-1}. As before this task can be accomplished by eliminating $\hat{\mathbf{p}}$ in order to obtain three nonlinear equations to be solved for $\hat{\mathbf{q}}$. The iterative solution procedure relies on Newton's method and makes use of the residual vector given by

$$\mathbf{R}(\hat{\mathbf{q}})=\begin{bmatrix} \frac{ml^2}{h_n}\left[-\frac{5}{2}q_1+\frac{3}{2}q_2+\frac{3}{2}q_3-\frac{1}{2}q_4\right]-p_1+h_n\int_0^1 \tilde{M}_1\partial_q V\,\mathrm{d}\alpha \\ \frac{ml^2}{h_n}[3q_1-3q_2-3q_3+3q_4]+h_n\int_0^1 \tilde{M}_2\partial_q V\,\mathrm{d}\alpha \\ \frac{ml^2}{h_n}[-7q_1+15q_2-12q_3+4q_4]-p_1+h_n\int_0^1 \tilde{M}_3\partial_q V\,\mathrm{d}\alpha \end{bmatrix}. \tag{40}$$

An outline of the iterative solution procedure for one typical time step is given in Table IV. The integrals in Table IV may be evaluated by employing Gaussian quadrature, analogous to (33).

4.3.2. *Numerical Accuracy and Quadrature*

In this section we examine the numerical accuracy of the time-stepping algorithms corresponding to cubic finite elements in time ($k=3$). In particular, we elaborate on the influence of the employed (Gaussian) quadrature rule on the numerical accuracy of the associated time-stepping scheme. As in Section 4.1.2, we calculate the mean-square norm of the error in the generalized displacements and momenta (see Eq. (36)). Again the time interval of interest is $I=[0, T]$, with $T=8$. The reference solution is obtained with linear elements incorporating exact quadrature and a time step size of $h_n=0.0001$.

The calculated results for the mean-square norm of the error in the generalized displacements are depicted in Fig. 8. Thus using only one Gauss point is not adequate for the cubic element since only second-order accuracy is attained. In contrast to that, the rate of convergence of the time-stepping scheme associated with two Gauss points is already four with respect to the mean-square norm. The application of three Gauss points improves the

TABLE IV

One-Dimensional Motion: Summary of Computations for One Typical Time Step in the Case $k = 3$ Corresponding to Cubic Finite Elements in Time

Given: initial conditions: $q_1,\ p_1$; time step size: h_n set iteration counter: $i = 1$

$$\mathbf{R}_1 = \frac{ml^2}{h_n}\begin{bmatrix} -\frac{5}{2}q_1 \\ 3q_1 \\ -7q_1 \end{bmatrix} + \begin{bmatrix} -p_1 \\ 0 \\ -p_1 \end{bmatrix}, \quad \mathbf{K}_1 = \frac{ml^2}{h_n}\begin{bmatrix} \frac{3}{2} & \frac{3}{2} & -\frac{1}{2} \\ -3 & -3 & 3 \\ 15 & -12 & 4 \end{bmatrix}$$

Find: vector of nodal unknowns

$$\hat{\mathbf{q}} = [q_2, q_3, q_4]^T \text{ and } \hat{\mathbf{p}} = [p_2,\ p_3,\ p_4]^T$$

(a) Compute residual

$$\mathbf{R}^{(i)} = \mathbf{R}_1 + \mathbf{K}_1\hat{\mathbf{q}}^{(i)} + \mathbf{R}_2^{(i)}$$

where

$$\mathbf{R}_2^{(i)} = \left[R_1^{(i)}, R_2^{(i)}, R_3^{(i)}\right]^T \text{ with } R_I^{(i)} = h_n \int_0^1 \tilde{M}_I\, \partial_q V^{(i)}\, \mathrm{d}\alpha$$

if $|\mathbf{R}^{(i)}| > \varepsilon$ goto (b) else goto (c)

(b) Compute tangent

$$\mathbf{K}^{(i)} = \mathbf{K}_1 + \mathbf{K}_2^{(i)}$$

where

$$\mathbf{K}_2^{(i)} = \begin{bmatrix} k_{11}^{(i)} & k_{12}^{(i)} & k_{13}^{(i)} \\ k_{21}^{(i)} & k_{22}^{(i)} & k_{23}^{(i)} \\ k_{31}^{(i)} & k_{32}^{(i)} & k_{33}^{(i)} \end{bmatrix} \text{ with } k_{IJ}^{(i)} = h_n \int_0^1 \tilde{M}_I M_{J+1} \partial_{qq}^2 V^{(i)}\, \mathrm{d}\alpha$$

Solve for increments

$$\Delta\hat{\mathbf{q}} = -{\mathbf{K}^{(i)}}^{-1}\mathbf{R}^{(i)}$$

Update generalized displacements

$$\hat{\mathbf{q}}^{(i+1)} = \hat{\mathbf{q}}^{(i)} + \Delta\hat{\mathbf{q}}$$

goto (a) with $i = i + 1$

(c) Update generalized momenta

$$p_2^{(i)} = \frac{ml^2}{h_n}\left[-\frac{175}{54}q_1 + \frac{13}{6}q_2^{(i)} + \frac{7}{6}q_3^{(i)} - \frac{5}{54}q_4^{(i)}\right] - \frac{11}{27}p_1$$

$$p_3^{(i)} = \frac{ml^2}{h_n}\left[\frac{74}{27}q_1 - \frac{20}{3}q_2^{(i)} + \frac{10}{3}q_3^{(i)} + \frac{16}{27}q_4^{(i)}\right] + \frac{11}{27}p_1$$

$$p_4^{(i)} = \frac{ml^2}{h_n}\left[-\frac{13}{2}q_1 + \frac{27}{2}q_2^{(i)} - \frac{27}{2}q_3^{(i)} + \frac{13}{2}q_4^{(i)}\right] - p_1$$

Note. Circular pendulum: $V = -mgl\cos q$.

accuracy slightly while retaining fourth-order accuracy. Increasing the order of the numerical quadrature by employing more than three Gauss points apparently does not improve the accuracy anymore.

4.3.3. *Energy Conservation and Quadrature*

In analogy to the case $k = 1$ in Section 4.1.3 we examine how the employed Gaussian quadrature influences the algorithmic energy conservation for $k = 3$. Referring to (20) exact quadrature implies exact energy conservation of the associated time-stepping scheme.

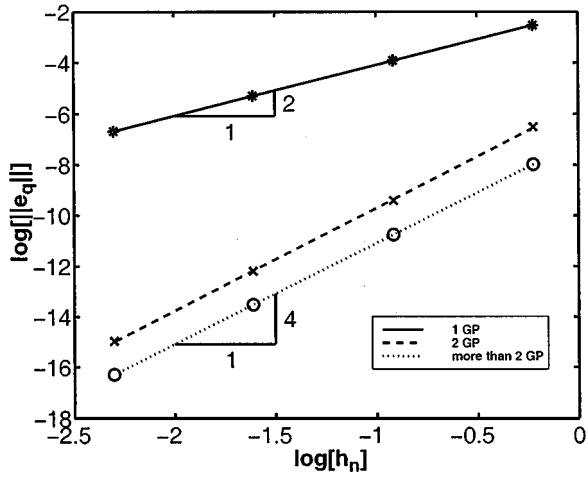

FIG. 8. Cubic elements ($k = 3$): Computed mean-square norm of the error in the generalized displacements for specific time-stepping schemes associated with different Gaussian quadrature rules.

Accordingly, with increasing order of the employed Gauss rule we expect a progressively energy conserving time-stepping algorithm. This feature is verified in view of Fig. 9, where again the error in the energy according to (37) is shown. Accordingly, already five Gauss points yield results being almost indistinguishable from exact quadrature.

5. SYSTEMS WITH SEVERAL DEGREES OF FREEDOM

Next, we discuss computational aspects related to the general case of conservative dynamical systems with several degrees of freedom. The corresponding numerical simulations deal with a planar double pendulum. The general time finite element method in (15) can be

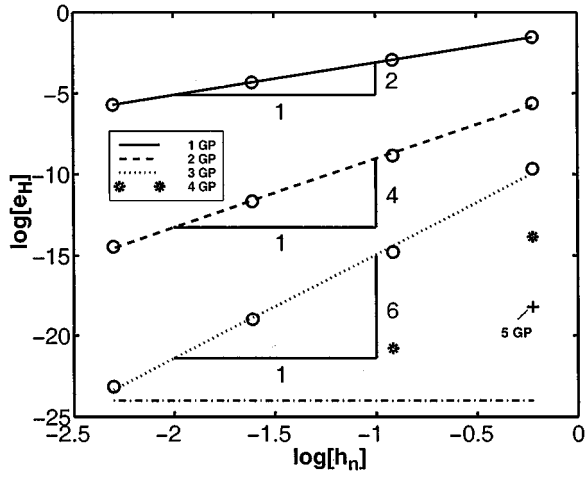

FIG. 9. Cubic elements ($k = 3$): Computed error in the total energy for different time-stepping schemes associated with specific quadrature rules.

recast in the form

$$\sum_{J=1}^{k} \int_0^1 \tilde{M}_I \tilde{M}_J \, \mathrm{d}\alpha \, \tilde{\mathbf{z}}_J - h_n \mathbb{J} \int_0^1 \tilde{M}_I \nabla H(\mathbf{z}) \, \mathrm{d}\alpha = \mathbf{0}, \tag{41}$$

for $I = 1, \ldots, k$. Here, use has been made of the matrices

$$\mathbf{z} = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix}$$

and

$$\mathbb{J} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix}, \tag{42}$$

where $\mathbf{0}$ and $\mathbf{I}$ are the $n_{\mathrm{dof}} \times n_{\mathrm{dof}}$ zero and identity matrices. Recall that the finite element interpolations in (10) imply the expression

$$\mathbf{z}(\alpha) = \sum_{I=1}^{k+1} M_I(\alpha) \mathbf{z}_I. \tag{43}$$

In general, exact calculation of the time integrals $\int_0^1 \tilde{M}_I \nabla H(\mathbf{z}) \, \mathrm{d}\alpha$ is not possible. On the other hand, referring to the treatment of algorithmic conservation properties in Section 3.2, in the autonomous case, algorithmic energy conservation is associated with the fulfillment of the relation

$$H(\mathbf{z}(1)) - H(\mathbf{z}(0)) = \sum_{I=1}^{k} \int_0^1 \tilde{M}_I \nabla H(\mathbf{z}) \, \mathrm{d}\alpha \cdot \tilde{\mathbf{z}}_I. \tag{44}$$

Application of standard quadrature formulas of the form

$$\int_0^1 \tilde{M}_I \nabla H(\mathbf{z}) \, \mathrm{d}\alpha \approx \sum_{l=1}^{N_l} \tilde{M}_I(\xi_l) \nabla H(\mathbf{z}(\xi_l)) w_l \tag{45}$$

will in general compromise the fulfillment of (44). The natural way to enforce algorithmic energy conservation is to increase the order of the integration rule. However, this can be prohibitively expensive. Alternatively, one can try to design reasonable nonstandard quadrature formulas which exactly fulfill Eq. (44) and thus lead to exactly energy conserving time-stepping schemes.

5.1. Nonstandard Quadrature Formula

The above-mentioned design of nonstandard quadrature rules is demonstrated next for $k = 1$, i.e., linear time finite elements. In this case Eq. (44) takes the form

$$H(\mathbf{z}(1)) - H(\mathbf{z}(0)) = \int_0^1 \nabla H(\mathbf{z}) \, \mathrm{d}\alpha \cdot \tilde{\mathbf{z}}, \tag{46}$$

with $\tilde{\mathbf{z}} = \mathbf{z}(1) - \mathbf{z}(0)$. Now consider the nonstandard quadrature formula

$$\int_0^1 \nabla H(\mathbf{z}) \, \mathrm{d}\alpha \approx \nabla H(\mathbf{z}(1/2)) + \frac{H(\mathbf{z}(1)) - H(\mathbf{z}(0)) - \nabla H(\mathbf{z}(1/2)) \cdot \tilde{\mathbf{z}}}{\|\tilde{\mathbf{z}}\|^2} \tilde{\mathbf{z}}. \tag{47}$$

It can be easily verified that the energy conservation condition (46) is fulfilled by applying formula (47). Hence the associated time-stepping scheme exactly conserves total energy. The specific form of (47) is in accordance with the discrete gradient method of Gonzalez [10]. There it has been shown that the discrete gradient appearing on the right-hand side of (47) is a second-order approximation to the exact gradient at the mid-point $\mathbf{z}(1/2)$. Accordingly, (47) can be interpreted as perturbation of the standard mid-point rule.

Indeed, our numerical investigations in the next section confirm that the time-stepping scheme resulting from the application of the nonstandard rule (47) retains the accuracy of the mid-point rule. In addition to that, exact algorithmic energy conservation has been observed.

5.2. Double Pendulum

The numerical example considered next deals with the planar double pendulum depicted in Fig. 10. The potential energy function can be written in the form

$$V(\mathbf{q}) = -Mgl[2\cos q_1 + \cos q_2], \tag{48}$$

and the kinetic energy is given by

$$T(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}} \cdot \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}, \tag{49}$$

with the generalized inertia matrix

$$\mathbf{M}(\mathbf{q}) = Ml^2 \begin{bmatrix} 2 & \cos(q_1 - q_2) \\ \cos(q_1 - q_2) & 1 \end{bmatrix}. \tag{50}$$

Accordingly, the generalized momentum is $\mathbf{p} = \partial_{\dot{\mathbf{q}}} T = M(\mathbf{q})\dot{\mathbf{q}}$, and the Hamiltonian function takes the form

$$H(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + \frac{1}{2}\mathbf{p} \cdot \mathbf{M}(\mathbf{q})^{-1}\mathbf{p}. \tag{51}$$

The implementation of the cg(1) method for the double pendulum at hand is summarized in Table V.

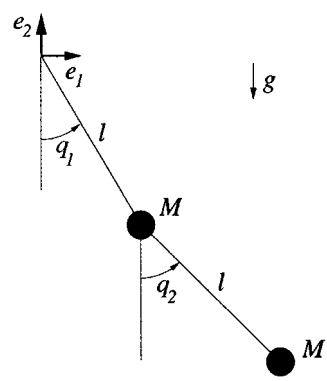

FIG. 10. Double pendulum.

TABLE V

Planar Double Pendulum: Summary of Computations for One Typical Time Step in the Case $k = 1$ Corresponding to Linear Time Finite Elements

Given: initial conditions: $\mathbf{z}_1 = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{p}_1 \end{bmatrix}$

time step size: h_n

set iteration counter: $i = 1$

$$\mathbb{J} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \quad \mathbb{I} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{0} & \mathbf{I} \end{bmatrix}$$

Find: nodal unknowns $\mathbf{z}_2 = \begin{bmatrix} \mathbf{q}_2 \\ \mathbf{p}_2 \end{bmatrix}$

(a) Compute residual

$$\mathbf{R}^{(i)} = \mathbf{z}_2^{(i)} - \mathbf{z}_1 - h_n \mathbb{J} \int_0^1 \nabla H(\mathbf{z}^{(i)})\, \mathrm{d}\alpha$$

if $\|\mathbf{R}^{(i)}\| > \varepsilon$ goto (b)

(b) Compute tangent

$$\mathbf{K}^{(i)} = \mathbb{I} - h_n \mathbb{J} \int_0^1 M_2 \nabla^2 H(\mathbf{z}^{(i)})\, \mathrm{d}\alpha$$

Update

$$\mathbf{z}_2^{(i+1)} = \mathbf{z}_2^{(i)} - \mathbf{K}^{(i)^{-1}} \mathbf{R}^{(i)}$$

goto (a) with $i = i + 1$

In the numerical example the parameters $M = 1$, $l = 1$, and $g = 9.81$ have been chosen. The initial conditions are

$$\mathbf{q}_0 = \begin{bmatrix} \pi/2 \\ \pi/2 \end{bmatrix}$$

and

$$\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{52}$$

The results obtained with a time step size of $h_n = 0.0001$ serve as reference solution, denoted as $\mathbf{z}_*(t)$. Figures 11 and 12 contain a sequence of deformed configurations for $0 \le t \le 1.51$ and $12.26 \le t \le 13.66$, respectively.

Figure 13 shows the computed error in the generalized displacements and momenta for linear time finite elements ($k = 1$). We calculated the norm of the error in z according to

$$\|\mathbf{e}_z\| = \left\{ \int_0^T \mathbf{e}_z \cdot \mathbb{M} \mathbf{e}_z \, \mathrm{d}t \right\}^{1/2}, \tag{53}$$

with $\mathbf{e}_z(t) = \mathbf{z}(t) - \mathbf{z}_*(t)$, scaling matrix $\mathbb{M}$ which has been set to $\mathbb{M} = \mathbb{I}$, the $2n_{\mathrm{dof}} \times 2n_{\mathrm{dof}}$ identity matrix, and $T = 1.5$. It can be seen that the nonstandard quadrature formula,

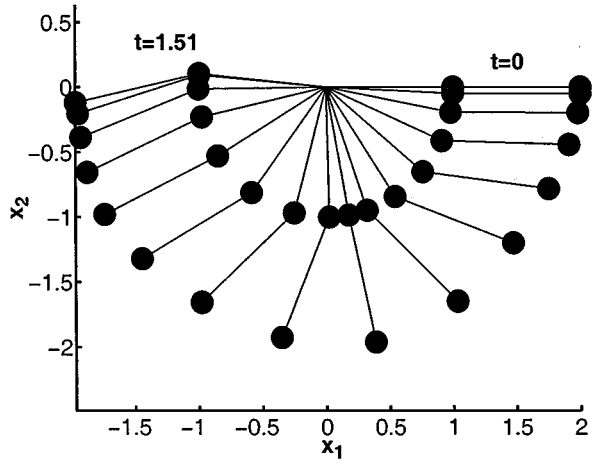

FIG. 11. Double pendulum: Sequence of deformed configurations for $0 \leq t \leq 1.51$.

Eq. (47), ("modified quadrature" in Fig. 13), yields the same order of accuracy as the mid-point rule. In this connection, using more than one Gauss point essentially yields the same results.

Figure 14 illustrates the enforcement of algorithmic energy conservation by increasing the order of the integration rule (for $h_n = 0.05$ and $k = 1$). Accordingly, applying 3 Gauss points prevents the blowup behavior exhibited by the mid-point rule (1 Gauss point), at least within the considered time interval $0 \leq t \leq 15$. Figure 14 can be interpreted as numerical verification of the inherent energy conservation property of the time finite element method under consideration.

Eventually, Fig. 15 shows the evolution of the y component of the position vector according to $y_1 = -l \cos q_1$.

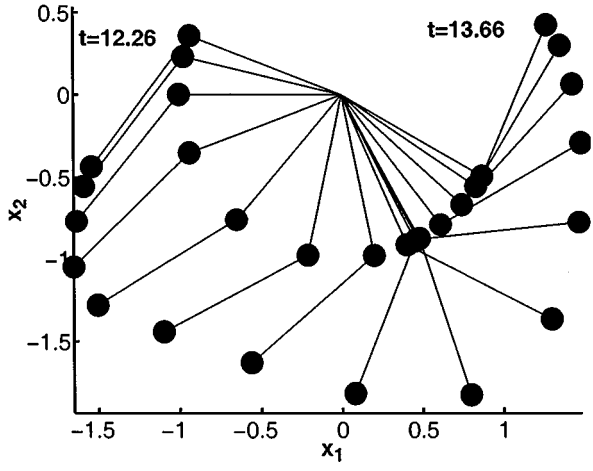

FIG. 12. Double pendulum: Sequence of deformed configurations for $12.26 \leq t \leq 13.66$.

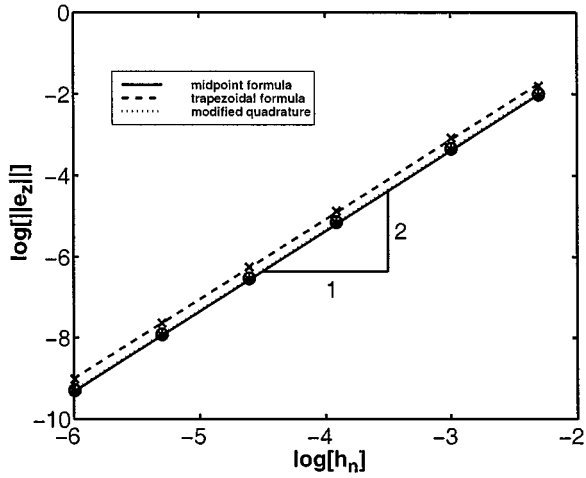

FIG. 13. Double pendulum: Computed norm of the error in $\mathbf{z}$ for linear time finite elements ($k = 1$).

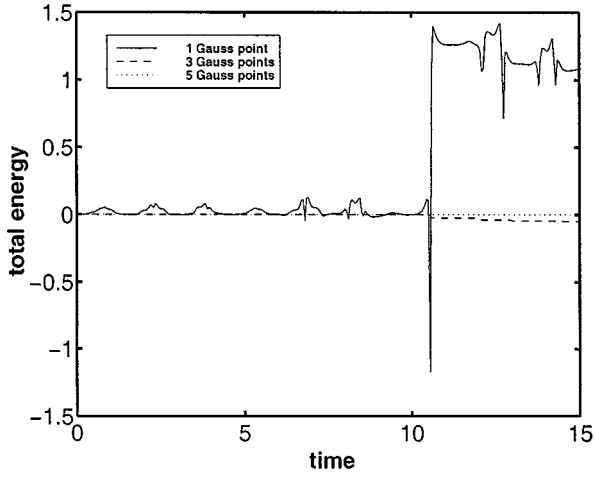

FIG. 14. Double pendulum: Increasing the number of Gauss points illustrates the inherent energy conservation property of the time finite element method under consideration ($h_n = 0.05$).

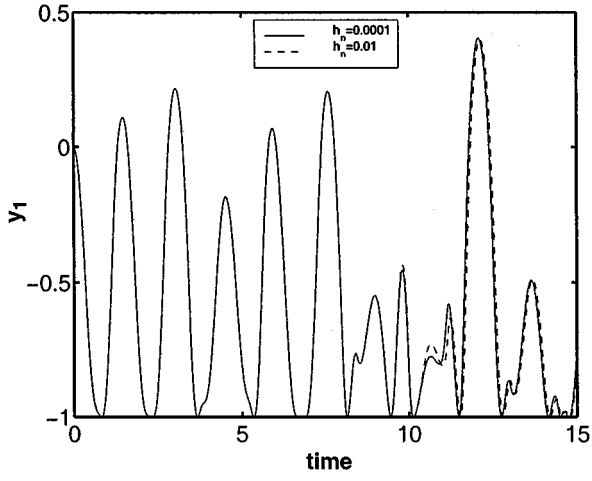

FIG. 15. Double pendulum: Component of position vector given by $y_1 = -l \cos q_1$.

6. CONCLUSIONS

We have derived a new family of time-stepping schemes for classical mechanics based upon the Petrov–Galerkin finite element approximation of Hamilton's canonical equations. Provided that a specific Hamiltonian function is given, the general time finite element formulation (15) can be used to generate particular time-stepping schemes. To this end, essentially two steps have to be performed:

1. The selection of the polynomial degree k of the Lagrangean shape functions specifies the corresponding finite element formulation.
2. A particular time-stepping scheme is finally obtained by the selection of the quadrature formula employed for the evaluation of the time integrals.

We saw that the time finite element method under consideration is inherently energy conserving. In the case of linear elements (i.e., $k = 1$) and one-dimensional motion the time integrals can be exactly calculated, thus leading to a time-stepping scheme which exactly conserves the total energy. Interestingly, this scheme has been shown to be identical with the difference method due to Greenspan [12].

In more general circumstances exact integration is rarely feasible. Therefore, we investigated how the application of standard quadrature rules affects the energy conservation property. Our numerical experiments related to $k = 1, 2, 3$ confirmed that algorithmic energy conservation can be simply achieved by increasing the order of the quadrature rule. On the other hand, in the general case of several degrees of freedom and $k = 1$ we have shown that exact algorithmic energy conservation can be maintained by applying a nonstandard quadrature formula. The design of this formula has been shown to be closely connected with the discrete gradient method proposed by Gonzalez [10]. In this respect it would be desirable to generalize the construction of nonstandard quadrature formulas for $k > 1$.

Despite the simplicity of the investigated model problem the system structure of the circular pendulum is similar to more involved problems such as N-body problems of classical mechanics or nonlinear elastodynamics in semidiscrete form. Accordingly, the time finite element approach advocated herein can be directly extended to dynamical systems where in many applications the algorithmic conservation of energy and angular momentum turns out to be especially important (see [4] and [5]).

APPENDIX: RELATION WITH GAUSS RUNGE–KUTTA METHODS

In this appendix we verify the connection of the proposed time finite element formulations with Gauss Runge–Kutta methods. According to Section 5, application of the time finite element method to the discretization of $\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z})$, with $\mathbf{f}(\mathbf{z}) = \mathbb{J}\nabla H(\mathbf{z})$, yields

$$\sum_{J=1}^{k+1} \int_0^1 \tilde{M}_I(\alpha) M'_J(\alpha)\, \mathrm{d}\alpha\, \mathbf{z}_J - h_n \int_0^1 \tilde{M}_I(\alpha) \mathbf{f}(\mathbf{z}(\alpha))\, \mathrm{d}\alpha = \mathbf{0}, \tag{A.1}$$

for $I = 1, \ldots, k$. Employing k-point Gaussian quadrature formulas the integrals in (A.1) are replaced by expressions of the form

$$\sum_{l=1}^{k} \tilde{M}_I(\xi_l) \mathbf{f}(\mathbf{z}(\xi_l)) w_l, \tag{A.2}$$

where the ξ_ls denote the location of the quadrature points and the w_ls are the associated weights. Then, taking into account (43), i.e., $\mathbf{z}(\alpha) = \sum_{I=1}^{k+1} M_I(\alpha)\mathbf{z}_I$ and the identity $\sum_{I=1}^{k} \tilde{M}_I = 1$, the system of equations in (A.1) can be recast in the form

$$\begin{aligned} \mathbf{z}(\xi_i) &= \mathbf{z}_1 + h_n \sum_{j=1}^{k} a_{ij}\mathbf{f}(\mathbf{z}(\xi_j)), \qquad i = 1, \ldots, k \\ \mathbf{z}_{k+1} &= \mathbf{z}_1 + h_n \sum_{i=1}^{k} w_i \mathbf{f}(\mathbf{z}(\xi_i)), \end{aligned} \tag{A.3}$$

which corresponds to a k-stage Runge–Kutta method (see, e.g., Stuart and Humphries [22]). The coefficients a_{ij} in (A.3) are determined by the finite element formulation. For instance, in the case of quadratic time finite elements ($k = 2$) dealt with in Section 4.2, the general method in (A.1) leads to

$$\begin{aligned} -5\mathbf{z}_1 + 4\mathbf{z}_2 + \mathbf{z}_3 - 6h_n \int_0^1 \tilde{M}_1(\alpha)\mathbf{f}(\mathbf{z}(\alpha))\, \mathrm{d}\alpha = \mathbf{0} \\ -\mathbf{z}_1 - 4\mathbf{z}_2 + 5\mathbf{z}_3 - 6h_n \int_0^1 \tilde{M}_2(\alpha)\mathbf{f}(\mathbf{z}(\alpha))\, \mathrm{d}\alpha = \mathbf{0}. \end{aligned} \tag{A.4}$$

A straightforward calculation yields

$$a_{ij} = \frac{w_j}{4}[M_2(\xi_i)\{5\tilde{M}_1(\xi_j) - \tilde{M}_2(\xi_j)\} + 4M_3(\xi_i)]. \tag{A.5}$$

Introducing the locations $\xi_1 = [1 - 1/\sqrt{3}]/2$ and $\xi_2 = [1 + 1/\sqrt{3}]/2$ of the quadrature points and values $w_1 = w_2 = 1/2$ of associated weights we obtain the coefficient matrix

$$[a_{ij}] = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \end{bmatrix}. \tag{A.6}$$

Accordingly, the coefficients ξ_i, a_{ij}, and w_i give rise to the Butcher tableau of the two-stage Gauss Runge–Kutta method (see, e.g., Hairer and Wanner [13]).

In the case of cubic time finite elements ($k = 3$) treated in section 4.3, the general method in (A.1) yields

$$\begin{aligned} -83\mathbf{z}_1 + 99\mathbf{z}_2 - 9\mathbf{z}_3 - 7\mathbf{z}_4 - 120h_n \int_0^1 \tilde{M}_1(\alpha)\mathbf{f}(\mathbf{z}(\alpha))\, \mathrm{d}\alpha = \mathbf{0} \\ -11\mathbf{z}_1 - 27\mathbf{z}_2 + 27\mathbf{z}_3 + 11\mathbf{z}_4 - 30h_n \int_0^1 \tilde{M}_2(\alpha)\mathbf{f}(\mathbf{z}(\alpha))\, \mathrm{d}\alpha = \mathbf{0} \\ 7\mathbf{z}_1 + 9\mathbf{z}_2 - 99\mathbf{z}_3 + 83\mathbf{z}_4 - 120h_n \int_0^1 \tilde{M}_3(\alpha)\mathbf{f}(\mathbf{z}(\alpha))\, \mathrm{d}\alpha = \mathbf{0}. \end{aligned} \tag{A.7}$$

Proceeding as before, a straightforward calculation renders the relation

$$a_{ij} = \frac{w_j}{27}[M_2(\xi_i)\{37\tilde{M}_1(\xi_j) + 4\tilde{M}_2(\xi_j) + \tilde{M}_3(\xi_j)\} + M_3(\xi_i)\{26\tilde{M}_1(\xi_j) + 23\tilde{M}_2(\xi_j) - 10\tilde{M}_3(\xi_j)\} + 27M_4(\xi_i)]. \tag{A.8}$$

Substitution of the abscissae and associated weights,

$$\begin{aligned} \xi_1 &= \frac{1}{2}\left[1 - \sqrt{\frac{3}{5}}\right] \qquad & w_1 &= \frac{5}{18} \\ \xi_2 &= \frac{1}{2} & w_2 &= \frac{4}{9} \\ \xi_3 &= \frac{1}{2}\left[1 + \sqrt{\frac{3}{5}}\right] & w_3 &= \frac{5}{18}, \end{aligned} \tag{A.9}$$

corresponding to the Gaussian quadrature rule for $k = 3$, yields the coefficient matrix

$$[a_{ij}] = \begin{bmatrix} \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\ \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\ \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \end{bmatrix}. \tag{A.10}$$

In summary, the coefficients ξ_i, a_{ij}, and w_i are the elements of the Butcher tableau associated with the three-stage Gauss Runge–Kutta method (see, e.g., Hairer and Wanner [13]).

REFERENCES

1. J. H. Argyris and D. W. Scharpf, Finite elements in time and space, *Nuclear Eng. Design* **10**, 456 (1969).
2. V. I. Arnold, *Mathematical Methods of Classical Mechanics* (Springer-Verlag, New York, 1989), 2nd ed.
3. C. D. Bailey, A new look at Hamilton's principle, *Found. Phys.* **5**(3), 433 (1975).
4. P. Betsch and P. Steinmann, Conservation properties of a time finite element method. I. Time-stepping schemes for N-body problems, *Int. J. Numer. Methods Eng.*, in press.
5. P. Betsch and P. Steinmann, Conservation properties of a time finite element method. II. Time-stepping schemes for nonlinear elastodynamics, Submitted for publication.
6. M. A. Crisfield and J. Shi, An energy conserving co-rotational procedure for non-linear dynamics with finite elements, *Nonlinear Dynam.* **9**, 37 (1996).
7. K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Computational Differential Equations*, Cambridge Univ. Press, Cambridge, UK (1996).
8. R. E. Gillilan and K. R. Wilson, Shadowing, rare events, and rubber bands. A variational Verlet algorithm for molecular dynamics, *J. Chem. Phys.* **97**(3), 1757 (1992).
9. H. Goldstein, *Classical Mechanics* (Addison-Wesley, Reading, MA, 1980), 2nd ed.
10. O. Gonzalez, Time integration and discrete Hamiltonian systems. *J. Nonlinear Sci.* **6**, 449 (1996).
11. O. Gonzalez and J. C. Simo, On the stability of symplectic and energy-momentum algorithms for non-linear Hamiltonian systems with symmetry. *Comput. Methods Appl. Mech. Eng.* **134**, 197 (1996).
12. D. Greenspan, Conservative numerical methods for $\ddot{x} = f(x)$. *J. Comput. Phys.* **56**, 28 (1984).
13. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations, II. Stiff and Differential-Algebraic Problems* (Springer-Verlag, New York, 1991).

14. T. J. R. Hughes, Stability, convergence and growth and decay of energy of the average acceleration method in nonlinear structural dynamics. *Comput. Struct.* **6**, 313 (1976).
15. T. J. R. Hughes, T. K. Caughey, and W. K. Liu, Finite-element methods for nonlinear elastodynamics which conserve energy. *J. Appl. Mech.* **45**, 366 (1978).
16. B. L. Hulme, One-step piecewise polynomial Galerkin methods for initial value problems. *Math. Comput.* **26**(118), 415 (1972).
17. D. Kuhl and E. Ramm, Constraint energy momentum algorithm and its application to nonlinear dynamics of shells. *Comput. Methods Appl. Mech. Eng.* **136**, 293 (1996).
18. R. A. LaBudde and D. Greenspan, Energy and momentum conserving methods of arbitrary order for the numerical integration of equations of motion. I. Motion of a single particle. *Numer. Math.* **25**, 323 (1976).
19. R. I. McLachlan, G. R. W. Quispel, and N. Robidoux, Unified approach to Hamiltonian systems, Poisson systems, gradient systems, and systems with Lyapunov functions or first integrals. *Phys. Rev. Lett.* **81**(12), 2399 (1998).
20. J. C. Simo and N. Tarnow, The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *J. Appl. Math. Phys. (ZAMP)* **43**, 757 (1992).
21. J. C. Simo, N. Tarnow, and K. K. Wong, Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Comput. Methods Appl. Mech. Eng.* **100**, 63 (1992).
22. A. M. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis* (Cambridge Univ. Press, Cambridge, UK, 1996).
23. J. M. Wendlandt and J. E. Marsden, Mechanical integrators derived from a discrete variational principle, *Physica D* **106**, 223 1997.
24. J. H. Williams, *Fundamentals of Applied Dynamics* (Wiley, New York, 1996).