

Research Article

An Energy Conservation Algorithm for Nonlinear Dynamic Equation

Jian Pang,¹ Yu Du,² Ping Hu,^{2,3} Weidong Li,^{2,3} and Z. D. Ma⁴

¹ Changan Auto Global R&D Center, State Key Laboratory of Vehicle NVH and Safety Technology, Chongqing 401120, China

² School of Automotive Engineering, Dalian University of Technology, Dalian 116024, China

³ State Key Laboratory of Structural Analysis of Industrial Equipment, Dalian University of Technology, Dalian 116024, China

⁴ College of Engineering, University of Michigan, Ann Arbor, MI 48109-2133, USA

Correspondence should be addressed to Weidong Li, liweidong@dlut.edu.cn

Received 14 July 2011; Revised 19 October 2011; Accepted 27 October 2011

Academic Editor: Ferenc Hartung

Copyright © 2012 Jian Pang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An energy conservation algorithm for numerically solving nonlinear multidegree-of-freedom (MDOF) dynamic equations is proposed. Firstly, by Taylor expansion and Duhamel integration, an integral iteration formula for numerically solving the nonlinear problems can be achieved. However, this formula still includes a parameter that is to be determined. Secondly, through some mathematical manipulations, the original dynamical equation can be further converted into an energy conservation equation which can then be used to determine the unknown parameter. Finally, an accurate numerical result for the nonlinear problem is achieved by substituting this parameter into the integral iteration formula. Several examples are used to compare the current method with the well-known Runge-Kutta method. They all show that the energy conservation algorithm introduced in this study can eliminate algorithm damping inherent in the Runge-Kutta algorithm and also has better stability for large integral steps.

1. Introduction

Numerical stability and algorithmic damping have been long recognized as two important aspects that need to be carefully handled in time integration algorithms for solving dynamic problems. Indeed, many works have been done in this area. For instance, to improve the stability, the Generalized- α method of Chung and Hulbert [1], the HHT- α method of Hilber et al. [2], and the WBZ- α method of Wood et al. [3] all demonstrate very good dissipation property either at low-frequency or high-frequency ranges. In references [4–8], Fung presents a series of time-step algorithms that are based on different mathematical and mechanical principles and can be used to deal with linear dynamical problems. Recent works on

numerical methods have been focusing on modeling long-term qualitative properties and stabilities in the numerical solutions of nonlinear dynamic problems. In the past, the feature of the energy conservation of a system has been widely used in various numerical integration methods to achieve satisfactory results. Since the property of exact energy conservation enables the numerical scheme to be stable without resorting to high-frequency numerical dissipation [9], the feature of the energy conservation appears particularly attractive in numerically solving nonlinear dynamic systems. Consequently, much effort has gone into the development of energy conserving time-stepping schemes. By adding an additional constraint through the energy conservation equation between adjacent time steps, Bui proposed a modified Newmark family for nonlinear dynamic analysis [10]. Similar works are also reported by LaBudde and Greenspan [11], Hughes et al. [12], Greenspan [13], Simo and Tarnow [14], Simo et al. [15], Greenspan [16], and Fung and Chow [17]. Interestingly, all works cited above derived their algorithms starting with the finite difference method, and most of them were based on Hamilton's canonical equations of motion.

Though energy conservative methods have showed some advantages, they also might suffer from some drawbacks in practical applications. In an attempt to obtain a stable large-step integration, Simo and Gonzalez [9] used the energy-momentum algorithm, which was obtained from the modification of midpoint scheme. By doing this, they risked to wrongly approximate slowly varying solution quantities in highly oscillatory systems, which was especially significant when fast and low modes are tightly coupled [18]. Moreover, with the addition of the constraint on the energy conservation, one has an overdetermined system that the exact solution has to satisfy. However, once the system is discretized and approximated in order to solve it numerically, the overdetermined system may not have a solution anymore. In other words, difficulties may occur in the multidimensional root finding of the corresponding nonlinear system of equations [19].

In this paper, the authors proposed a time integration formula and scheme which can maintain the system energy conservation constraint automatically. However, this method is also different from the energy conservative methods developed previously in the following four aspects. Firstly, the solution of the nonlinear dynamic equation was presented analytically by Duhamel integral in the current method. Secondly, the current method used the Taylor expansion to approximate the exact solution of the nonlinear equation. During this process, an undetermined parameter was introduced. Thirdly, substituting the approximate solution into the analytical solution, an iterative formula with the undetermined parameter was derived. Finally, the energy conservation equation was established and the undetermined parameter was obtained. In practice, how to introduce and obtain the undetermined parameter should be carefully considered case by case, especially when the right hand term of the nonlinear equation includes functions of velocity and time. Another merit of the current algorithm was that it behaved very stable under large time steps. Comparisons with the Runge-Kutta method showed that the proposed method had much better stabilities in solving different types of nonlinear equations.

The remainder of this paper is arranged as follows. In Section 2, a detailed process of deriving the integral iteration formula is presented. In the iteration formula, we will show that how an undetermined parameter is introduced into the formula. In Section 3, the energy conservation equation is established for obtaining the algebraic expression of the undetermined parameter. Section 4 focuses on the calculation of the multinomial interpolation used in Section 2. Section 5 shows some representative numerical examples which compare the current method with the popular Runge-Kutta method in terms of

algorithmic damping and stability. Finally, some discussions and conclusions are given in Section 6.

2. Derivation of the Integral Iteration Formula

Step-by-step time-integration algorithms are commonly used to solve dynamic equations which mostly come from actual engineering problems. By spatial discretization using the finite element method, a nonlinear system may be represented by a second order nonlinear ordinary differential equation as

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, t), \quad (2.1)$$

where \mathbf{M} , \mathbf{K} are n -by- n constant mass and stiffness matrices, respectively. \mathbf{x} , $\dot{\mathbf{x}}$, $\ddot{\mathbf{x}}$, \mathbf{f} are vectors with rank n representing the displacement, velocity, and acceleration, respectively. t is time. The right hand term in (2.1), $\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, t)$, is the force vector that includes all external forces such as the damping forces and the nonlinear forces. Using the matrix decomposition, the mass matrix \mathbf{M} can be expressed as

$$\mathbf{M} = \mathbf{L} \cdot \mathbf{L}^T \quad \mathbf{M}^{-1} = \mathbf{L}^{-T} \cdot \mathbf{L}^{-1}. \quad (2.2)$$

Substituting (2.2) into (2.1) and multiplying by \mathbf{L}^{-1} , we can obtain

$$\mathbf{L}^T \ddot{\mathbf{x}} + \mathbf{L}^{-1} \mathbf{K} \mathbf{L}^{-T} \cdot \mathbf{L}^T \mathbf{x} = \mathbf{L}^{-1} \mathbf{f}. \quad (2.3)$$

Knowing that \mathbf{L} , \mathbf{L}^{-1} , \mathbf{L}^T , \mathbf{L}^{-T} are all constant matrices, a variable substitution can be executed as

$$\mathbf{y} = \mathbf{L}^T \mathbf{x}, \quad \dot{\mathbf{y}} = \mathbf{L}^T \dot{\mathbf{x}}, \quad \ddot{\mathbf{y}} = \mathbf{L}^T \ddot{\mathbf{x}}. \quad (2.4)$$

Substituting (2.4) into (2.3), a new dynamic system which is equivalent to the original system can be obtained as

$$\ddot{\mathbf{y}} + \bar{\mathbf{K}}\mathbf{y} = \mathbf{F}(t), \quad \mathbf{F}(t) = \mathbf{L}^{-1} \mathbf{f}(t), \quad \bar{\mathbf{K}} = \mathbf{L}^{-1} \mathbf{K} \mathbf{L}^{-T}. \quad (2.5)$$

Let

$$\bar{\mathbf{K}} = \mathbf{K}_0 + \mathbf{D}, \quad (2.6)$$

where \mathbf{D} is a diagonal matrix and its diagonal elements are the diagonal elements of matrix $\bar{\mathbf{K}}$. \mathbf{K}_0 is a matrix whose diagonal elements are zero, and other elements are equal to those in $\bar{\mathbf{K}}$. Note that $D_{ii} \geq 0$ for \mathbf{M} which is a positive definite matrix and \mathbf{K} is a semipositive definite matrix. Let

$$D_{ii} = d_i^2. \quad (2.7)$$

Then (2.5) can be rewritten as

$$\dot{\mathbf{y}} + \mathbf{D}\mathbf{y} = \mathbf{F}(t) - \mathbf{K}_0\mathbf{y}. \quad (2.8)$$

The separate form of the above matrix equation can be expressed as follows:

$$\ddot{y}_i + d_i^2 y_i = F_i(t) - \mathbf{k}_0^i \mathbf{y}, \quad i = 1, 2, \dots, n, \quad (2.9)$$

where \mathbf{k}_0^i is the i th row of the matrix \mathbf{K}_0 . From (2.9), it is easy to see that the analytical solution of the displacement and the velocity can be obtained by the Duhamel integral as

$$\begin{aligned} y_i(t) = & y_i(t_k) \cos d_i(t - t_k) + \dot{y}_i(t_k) \frac{\sin d_i(t - t_k)}{d_i} \\ & + \frac{1}{d_i} \int_{t_k}^t F_i(\zeta) \sin d_i(t - \zeta) d\zeta - \frac{1}{d_i} \int_{t_k}^t \mathbf{k}_0^i \mathbf{y}(\zeta) \sin d_i(t - \zeta) d\zeta, \end{aligned} \quad (2.10)$$

$$\begin{aligned} \dot{y}_i(t) = & -y_i(t_k) d_i \sin d_i(t - t_k) + \dot{y}_i(t_k) \cos d_i(t - t_k) \\ & + \int_{t_k}^t F_i(\zeta) \cos d_i(t - \zeta) d\zeta - \int_{t_k}^t \mathbf{k}_0^i \mathbf{y}(\zeta) \cos d_i(t - \zeta) d\zeta. \end{aligned} \quad (2.11)$$

In order to derive the time integral formula, let $t = t_k + \tau$ in (2.10) and (2.11), where τ is the integral time step, then we have

$$\begin{aligned} y_i(t_k + \tau) = & y_i(t_k) \cos d_i \tau + \dot{y}_i(t_k) \frac{\sin d_i \tau}{d_i} + \frac{1}{d_i} \int_{t_k}^{t_k + \tau} F_i(\zeta) \sin d_i(t_k + \tau - \zeta) d\zeta \\ & - \frac{1}{d_i} \int_{t_k}^{t_k + \tau} \mathbf{k}_0^i \mathbf{y}(\zeta) \sin d_i(t_k + \tau - \zeta) d\zeta, \end{aligned} \quad (2.12)$$

$$\begin{aligned} \dot{y}_i(t_k + \tau) = & -y_i(t_k) d_i \sin d_i \tau + \dot{y}_i(t_k) \cos d_i \tau + \int_{t_k}^{t_k + \tau} F_i(\zeta) \cos d_i(t_k + \tau - \zeta) d\zeta \\ & - \int_{t_k}^{t_k + \tau} \mathbf{k}_0^i \mathbf{y}(\zeta) \cos d_i(t_k + \tau - \zeta) d\zeta. \end{aligned} \quad (2.13)$$

Inspecting (2.12) and (2.13), it can be found that there are still some unknown parameters that need to be identified before the time integral formula can be carried out numerically. They are the right hand side terms consisting of the undetermined variables $\mathbf{y}(t)$ and $F_i(t)$. The latter one may also be a function of $\mathbf{y}(t)$ and $\dot{\mathbf{y}}(t)$. To proceed, the Taylor expansion formula is used to expand $\mathbf{y}(t)$ on the interval $(t_k \leq t \leq t_k + \tau)$ as

$$\mathbf{y}(t) = \mathbf{y}(t_k) + (t - t_k) \dot{\mathbf{y}}(t_k) + \frac{(t - t_k)^2}{2} \ddot{\mathbf{y}}(t_k) + \frac{(t - t_k)^3}{6} \mathbf{a}. \quad (2.14)$$

In (2.14), $\mathbf{y}(t)$ is expanded to the third order term following exactly the Taylor expansion process, while in calculating the fourth order term, a new variable vector \mathbf{a} is

introduced. It should be pointed out that although the Taylor expansion is an approximation to the original variable $\mathbf{y}(t)$, (2.14) is still an exact expression of the variable $\mathbf{y}(t)$. This is because the last term in (2.14) which includes the newly introduced vector \mathbf{a} can be interpreted to compensate for the difference between $\mathbf{y}(t)$ and the summation of the first three terms in (2.14). In order to express the vector \mathbf{a} by the variable $\mathbf{y}(t)$, let $\mathbf{y}_1 = \mathbf{y}(t = t_k + \tau)$, $\mathbf{y}_0 = \mathbf{y}(t = t_k)$. From (2.14), it can be obtained that

$$\mathbf{a} = \frac{6}{\tau^3} \left(\mathbf{y}_1 - \mathbf{y}_0 - \tau \dot{\mathbf{y}}_0 - \frac{\tau^2}{2} \ddot{\mathbf{y}}_0 \right), \quad (2.15)$$

$$\mathbf{y}(t) = N_1(t)\mathbf{y}_0 + N_2(t)\dot{\mathbf{y}}_0 + N_3(t)\ddot{\mathbf{y}}_0 + N_4(t)\mathbf{y}_1,$$

where:

$$\begin{aligned} N_1(t) &= 1 - \frac{(t - t_k)^3}{\tau^3}, & N_2(t) &= t - t_k - \frac{(t - t_k)^3}{\tau^2}, \\ N_3(t) &= \frac{(t - t_k)^2}{2} - \frac{(t - t_k)^3}{2\tau}, & N_4(t) &= \beta(t - t_k)^3(t - t_k - \tau) + \frac{(t - t_k)^3}{\tau^3}, \end{aligned} \quad (2.16)$$

where an undetermined parameter β has been introduced in $N_4(t)$ to regulate the stability of the algorithm and will be determined by the energy conservation equation in the next section. By $t = t_k$, $t = t_k + \tau$ in (2.14), we can obtain

$$\mathbf{y}(t_k) = \mathbf{y}_0, \quad \dot{\mathbf{y}}(t_k) = \dot{\mathbf{y}}_0, \quad \ddot{\mathbf{y}}(t_k) = \ddot{\mathbf{y}}_0, \quad \mathbf{y}(t_k + \tau) = \mathbf{y}_1. \quad (2.17)$$

By means of multinomial interpolation, $\mathbf{f}(t)$ ($t_k \leq t \leq t_k + \tau$) can be written as

$$\mathbf{f}(t) = \mathbf{r}_0 + (t - t_k)\mathbf{r}_1 + (t - t_k)^2\mathbf{r}_2 + (t - t_k)^3\mathbf{r}_3 + o((t - t_k)^4). \quad (2.18)$$

Here we use the third order interpolation, and generally one can choose the order of interpolation discretionarily based on solely the algorithm accuracy order that is needed. Different interpolation order will lead to different integration formulas. Now, we will derive the integration formulas first, and the discussion of the interpolation will be addressed at Section 4. Using (2.5) and (2.14), the last two terms of the right hand of (2.11) can be expressed separately as

$$\begin{aligned} \frac{1}{d_i} \int_{t_k}^{t_k + \tau} \mathbf{k}_0^i \mathbf{y}(\zeta) \sin d_i(t_k + \tau - \zeta) d\zeta &= \alpha_0^i \mathbf{k}_0^i \mathbf{y}_0 + \alpha_1^i \mathbf{k}_0^i \dot{\mathbf{y}}_0 + \alpha_2^i \mathbf{k}_0^i \ddot{\mathbf{y}}_0 + \alpha_3^i \mathbf{k}_0^i \mathbf{y}_1, \\ \frac{1}{d_i} \int_{t_k}^{t_k + \tau} \mathbf{L}_i^{-1} \mathbf{f}(\zeta) \sin d_i(t_k + \tau - \zeta) d\zeta &= \gamma_0^i \mathbf{L}_i^{-1} \mathbf{r}_0 + \gamma_1^i \mathbf{L}_i^{-1} \mathbf{r}_1 + \gamma_2^i \mathbf{L}_i^{-1} \mathbf{r}_2 + \gamma_3^i \mathbf{L}_i^{-1} \mathbf{r}_3, \end{aligned} \quad (2.19)$$

where $\alpha_k^i, \gamma_k^i, k = 0, 1, 2, 3$ are scalar and can be obtained by follow polynomials

$$\begin{aligned}
 \alpha_0^i &= -\frac{(6 \sin d_i \tau + d_i^3 \tau^3 \cos d_i \tau - 6 d_i \tau)}{d_i^5 \tau^3}, \\
 \alpha_1^i &= \frac{(6 d_i \tau - d_i^2 \tau^2 \sin d_i \tau - 6 \sin d_i \tau)}{d_i^5 \tau^2}, \\
 \alpha_2^i &= \frac{(2 d_i \tau - 3 \sin d_i \tau + d_i \tau \cos d_i \tau)}{d_i^5 \tau}, \\
 \alpha_3^i &= \frac{(6 \sin d_i \tau + d_i^3 \tau^3 - 6 d_i \tau)}{d_i^5 \tau^3} - \frac{\beta(d_i^2 \tau^2 - 4 + 4 \cos d_i \tau + d_i \tau \sin d_i \tau)}{d_i^6 \tau^4}, \\
 \gamma_0^i &= \frac{(1 - \cos d_i \tau)}{d_i^2}, \\
 \gamma_1^i &= \frac{(d_i \tau - \sin d_i \tau)}{d_i^3}, \\
 \gamma_2^i &= \frac{(d_i^2 \tau^2 - 2 + 2 \cos d_i \tau)}{d_i^4}, \\
 \gamma_3^i &= \frac{(d_i^3 \tau^3 - 6 d_i \tau + 6 \sin d_i \tau)}{d_i^5}.
 \end{aligned} \tag{2.20}$$

Substituting (2.20) into (2.12), we have

$$\begin{aligned}
 y_i(t_k + \tau) &= y_i(t_k) \cos d_i \tau + \dot{y}_i(t_k) \frac{\sin d_i \tau}{d_i} + \gamma_0^i \mathbf{L}_i^{-1} \mathbf{r}_0 + \gamma_1^i \mathbf{L}_i^{-1} \mathbf{r}_1 + \gamma_2^i \mathbf{L}_i^{-1} \mathbf{r}_2 \\
 &+ \gamma_3^i \mathbf{L}_i^{-1} \mathbf{r}_3 - \left(\alpha_0^i \mathbf{K}_0^i \mathbf{y}_0 + \alpha_1^i \mathbf{K}_0^i \dot{\mathbf{y}}_0 + \alpha_2^i \mathbf{K}_0^i \ddot{\mathbf{y}}_0 + \alpha_3^i \mathbf{K}_0^i \mathbf{y}_1 \right) \quad (i = 1, 2, \dots, n).
 \end{aligned} \tag{2.21}$$

Knowing that $\mathbf{y}_1 = \mathbf{y}(t = t_k + \tau)$, so we can write the above equation in a matrix form:

$$\begin{aligned}
 \mathbf{y}_1 &= \mathbf{U}_0 \mathbf{y}_0 + \mathbf{U}_1 \dot{\mathbf{y}}_0 + \gamma_0 \mathbf{L}^{-1} \mathbf{r}_0 + \gamma_1 \mathbf{L}^{-1} \mathbf{r}_1 + \gamma_2 \mathbf{L}^{-1} \mathbf{r}_2 \\
 &+ \gamma_3 \mathbf{L}^{-1} \mathbf{r}_3 - \alpha_0 \mathbf{K}_0 \mathbf{y}_0 - \alpha_1 \mathbf{K}_0 \dot{\mathbf{y}}_0 - \alpha_2 \mathbf{K}_0 \ddot{\mathbf{y}}_0 - \alpha_3 \mathbf{K}_0 \mathbf{y}_1,
 \end{aligned} \tag{2.22}$$

where $\mathbf{U}_0, \mathbf{U}_1, \alpha_k, \gamma_k$ ($k = 0, 1, 2, 3$) are diagonal matrices and their diagonal elements are $\cos d_i \tau, \sin d_i \tau / d_i, \alpha_k^i, \gamma_k^i$ ($i = 1, 2, \dots, n, k = 0, 1, 2, 3$), respectively. From (2.22), the iterative solution \mathbf{y}_1 can be expressed as

$$\begin{aligned}
 (\mathbf{I} + \alpha_3 \mathbf{K}_0) \mathbf{y}_1 &= (\mathbf{U}_0 - \alpha_0 \mathbf{K}_0) \mathbf{y}_0 + (\mathbf{U}_1 - \alpha_1 \mathbf{K}_0) \dot{\mathbf{y}}_0 - \alpha_2 \mathbf{K}_0 \ddot{\mathbf{y}}_0 + \gamma_0 \mathbf{L}^{-1} \mathbf{r}_0 \\
 &+ \gamma_1 \mathbf{L}^{-1} \mathbf{r}_1 + \gamma_2 \mathbf{L}^{-1} \mathbf{r}_2 + \gamma_3 \mathbf{L}^{-1} \mathbf{r}_3.
 \end{aligned} \tag{2.23}$$

Multiplying (2.23) by $\mathbf{L}\boldsymbol{\alpha}_3^{-1}$, we have

$$\begin{aligned} \mathbf{L}(\boldsymbol{\alpha}_3^{-1} + \mathbf{K}_0)\mathbf{y}_1 &= \mathbf{L}\boldsymbol{\alpha}_3^{-1}(\mathbf{U}_0 - \boldsymbol{\alpha}_0\mathbf{K}_0)\mathbf{y}_0 + \mathbf{L}\boldsymbol{\alpha}_3^{-1}(\mathbf{U}_1 - \boldsymbol{\alpha}_1\mathbf{K}_0)\dot{\mathbf{y}}_0 \\ &\quad - \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_2\mathbf{K}_0\ddot{\mathbf{y}}_0 + \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_0\mathbf{L}^{-1}\mathbf{r}_0 + \boldsymbol{\gamma}_1\mathbf{L}^{-1}\mathbf{r}_1 + \boldsymbol{\gamma}_2\mathbf{L}^{-1}\mathbf{r}_2 + \boldsymbol{\gamma}_3\mathbf{L}^{-1}\mathbf{r}_3. \end{aligned} \quad (2.24)$$

Note that \mathbf{K}_0 is a matrix whose diagonal elements are zero, and other elements are equal to those in $\bar{\mathbf{K}}$ and $\boldsymbol{\alpha}_k$ is a diagonal matrix, so $\mathbf{L}(\boldsymbol{\alpha}_3^{-1} + \mathbf{K}_0)$ is a symmetrical matrix. The left hand side of (2.24) can be written as

$$\mathbf{L}(\boldsymbol{\alpha}_3^{-1} + \mathbf{K}_0)\mathbf{L}^T\mathbf{L}^{-T}\mathbf{y}_1 = (\mathbf{L}\boldsymbol{\alpha}_3^{-1}\mathbf{L}^T + \mathbf{L}\mathbf{K}_0\mathbf{L}^T)\mathbf{L}^{-T}\mathbf{y}_1. \quad (2.25)$$

From (2.5) and (2.6), we have

$$\mathbf{L}\mathbf{K}_0\mathbf{L}^T = \mathbf{L} \cdot (\bar{\mathbf{K}} - \mathbf{D}) \cdot \mathbf{L}^T = \mathbf{L} \cdot \mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-T} \cdot \mathbf{L}^T - \mathbf{L}\mathbf{D}\mathbf{L}^T = \mathbf{K} - \mathbf{L}\mathbf{D}\mathbf{L}^T. \quad (2.26)$$

From (2.3) and (2.26), the left hand of (2.24) can be written as

$$\mathbf{L}(\boldsymbol{\alpha}_3^{-1} + \mathbf{K}_0)\mathbf{y}_1 = [\mathbf{K} + \mathbf{L}(\boldsymbol{\alpha}_3^{-1} - \mathbf{D})\mathbf{L}^T] \cdot \mathbf{x}_1. \quad (2.27)$$

All the terms in the right hand side of (2.24) can be expressed separately as

$$\begin{aligned} \mathbf{L}\boldsymbol{\alpha}_3^{-1}(\mathbf{U}_0 - \boldsymbol{\alpha}_0\mathbf{K}_0)\mathbf{y}_0 &= [\mathbf{L}(\boldsymbol{\alpha}_3^{-1}\mathbf{U}_0 + \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_0\mathbf{D})\mathbf{L}^T - \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_0\mathbf{L}^{-1}\mathbf{K}] \cdot \mathbf{x}_0, \\ \mathbf{L}\boldsymbol{\alpha}_3^{-1}(\mathbf{U}_1 - \boldsymbol{\alpha}_1\mathbf{K}_0)\dot{\mathbf{y}}_0 &= [\mathbf{L}(\boldsymbol{\alpha}_3^{-1}\mathbf{U}_1 + \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_1\mathbf{D})\mathbf{L}^T - \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_1\mathbf{L}^{-1}\mathbf{K}] \cdot \dot{\mathbf{x}}_0, \\ \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_2\mathbf{K}_0\ddot{\mathbf{y}}_0 &= \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_2\mathbf{L}^{-1}\mathbf{L}\mathbf{K}^0\mathbf{L}^T\mathbf{L}^{-T}\ddot{\mathbf{y}}_0 = \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_2\mathbf{L}^{-1}(\mathbf{K} - \mathbf{L}\mathbf{D}\mathbf{L}^T) \cdot \ddot{\mathbf{x}}_0. \end{aligned} \quad (2.28)$$

After substituting proper variables, the iteration formula in terms of the original variables can be expressed as

$$\begin{aligned} &[\mathbf{K} + \mathbf{L}(\boldsymbol{\alpha}_3^{-1} - \mathbf{D})\mathbf{L}^T] \cdot \mathbf{x}_1 \\ &= [\mathbf{L}(\boldsymbol{\alpha}_3^{-1}\mathbf{U}_0 + \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_0\mathbf{D})\mathbf{L}^T - \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_0\mathbf{L}^{-1}\mathbf{K}] \cdot \mathbf{x}_0 \\ &\quad + [\mathbf{L}(\boldsymbol{\alpha}_3^{-1}\mathbf{U}_1 + \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_1\mathbf{D})\mathbf{L}^T - \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_1\mathbf{L}^{-1}\mathbf{K}] \cdot \dot{\mathbf{x}}_0 - \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_2\mathbf{L}^{-1}(\mathbf{K} - \mathbf{L}\mathbf{D}\mathbf{L}^T) \cdot \ddot{\mathbf{x}}_0 \\ &\quad + \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_0\mathbf{L}^{-1}\mathbf{r}_0 + \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_1\mathbf{L}^{-1}\mathbf{r}_1 + \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_2\mathbf{L}^{-1}\mathbf{r}_2 + \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_3\mathbf{L}^{-1}\mathbf{r}_3. \end{aligned} \quad (2.29)$$

In (2.29), there is a term consisting of a double derivative. According to (2.3), the term with the double derivative $\ddot{\mathbf{x}}_0$ can be replaced by $-\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{K}\mathbf{x}_0 + \mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{f}_0$. Finally, the displacement iteration formula can be obtained as

$$\begin{aligned}
& \left[\mathbf{K} + \mathbf{L}(\boldsymbol{\alpha}_3^{-1} - \mathbf{D})\mathbf{L}^T \right] \cdot \mathbf{x}_1 \\
&= \mathbf{L} \cdot \left[(\boldsymbol{\alpha}_3^{-1}\mathbf{U}_0 + \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_0\mathbf{D})\mathbf{L}^T - \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_0\mathbf{L}^{-1}\mathbf{K} + \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_2\mathbf{L}^{-1}(\mathbf{K} - \mathbf{L}\mathbf{D}\mathbf{L}^T)\mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{K} \right] \cdot \mathbf{x}_0 \\
&+ \mathbf{L} \cdot \left[(\boldsymbol{\alpha}_3^{-1}\mathbf{U}_1 + \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_1\mathbf{D})\mathbf{L}^T - \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_1\mathbf{L}^{-1}\mathbf{K} \right] \cdot \dot{\mathbf{x}}_0 \\
&+ \mathbf{L} \cdot \left[\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_0\mathbf{L}^{-1} - \boldsymbol{\alpha}_3^{-1}\boldsymbol{\alpha}_2\mathbf{L}^{-1}(\mathbf{K} - \mathbf{L}\mathbf{D}\mathbf{L}^T) \right] \cdot \mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{r}_0 \\
&+ \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_1\mathbf{L}^{-1}\mathbf{r}_1 + \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_2\mathbf{L}^{-1}\mathbf{r}_2 + \mathbf{L}\boldsymbol{\alpha}_3^{-1}\boldsymbol{\gamma}_3\mathbf{L}^{-1}\mathbf{r}_3.
\end{aligned} \tag{2.30}$$

Substituting (2.14) and (2.15) into (2.13) and through some mathematical manipulations, the velocity iteration formula can be obtained as

$$\begin{aligned}
\dot{\mathbf{x}}_1 &= \mathbf{R}_\tau\mathbf{x}_1 + \mathbf{R}_0\mathbf{x}_0 + \mathbf{R}_1\dot{\mathbf{x}}_0 + \mathbf{L}^{-T} \left[\boldsymbol{\eta}_0\mathbf{L}^{-1} - (\mathbf{c}_2\mathbf{L}^{-1}\mathbf{K} - \mathbf{c}_2\mathbf{D}\mathbf{L}^T)\mathbf{L}^{-T}\mathbf{L}^{-1} \right] \mathbf{r}_0 \\
&+ \mathbf{L}^{-T}\boldsymbol{\eta}_1\mathbf{L}^{-1}\mathbf{r}_1 + \mathbf{L}^{-T}\boldsymbol{\eta}_2\mathbf{L}^{-1}\mathbf{r}_2 + \mathbf{L}^{-T}\boldsymbol{\eta}_3\mathbf{L}^{-1}\mathbf{r}_3,
\end{aligned} \tag{2.31}$$

where $\boldsymbol{\eta}_0, \boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \boldsymbol{\eta}_3$ are diagonal matrices with diagonal elements $\eta_{l_0}^i, \eta_{l_1}^i, \eta_{l_2}^i, \eta_{l_3}^i$, respectively. \mathbf{c}_m is also a diagonal matrix and its diagonal elements are c_m^i ($m = 0, 1, 2, 3$). Every term in the right hand side of (2.31) is given as bellow:

$$\begin{aligned}
\mathbf{R}_0 &= \mathbf{L}^{-T} \left[(\mathbf{V}_0\mathbf{L}^T + \mathbf{c}_0\mathbf{D}\mathbf{L}^T - \mathbf{c}_0\mathbf{L}^{-1}\mathbf{K}) + \mathbf{c}_2\mathbf{L}^{-1}\mathbf{K} - \mathbf{c}_2\mathbf{D}\mathbf{L}^T \right] \mathbf{L}^{-T}\mathbf{L}^{-1} \cdot \mathbf{K}, \\
\mathbf{R}_1 &= \mathbf{L}^{-T} (\mathbf{V}_1\mathbf{L}^T + \mathbf{c}_1\mathbf{D}\mathbf{L}^T - \mathbf{c}_1\mathbf{L}^{-1}\mathbf{K}), \\
\mathbf{R}_\tau &= -\mathbf{L}^{-T} \mathbf{c}_3 (\mathbf{L}^{-1}\mathbf{K} - \mathbf{D}\mathbf{L}^T), \\
c_0^i &= \frac{(d_i^3\tau^3 \sin d_i\tau - 6 \cos d_i\tau - 3d_i^2\tau^2 + 6)}{d_i^4\tau^3}, \\
c_1^i &= -\frac{(d_i^2\tau^2 \cos d_i\tau + 6 \cos d_i\tau + 2d_i^2\tau^2 - 6)}{d_i^4\tau^2}, \\
c_2^i &= -\frac{(1/2)(2d_i\tau \sin d_i\tau + 6 \cos d_i\tau + d_i^2\tau^2 - 6)}{d_i^4\tau}, \\
c_3^i &= \frac{(6 \cos d_i\tau + 3d_i^2\tau^2 - 6)}{d_i^4\tau^3} - \frac{\beta(6d_i\tau \cos d_i\tau - 24 \sin d_i\tau - d_i^2\tau^2 + 18)}{d_i^5\tau^4}, \\
\eta_0^i &= \frac{\sin d_i\tau}{d_i} \quad \eta_1^i = \frac{1 - \cos d_i\tau}{d_i^2} \quad \eta_2^i = \frac{2d_i\tau - 2 \sin d_i\tau}{d_i^3}, \\
\eta_3^i &= \int_{t_k}^{t_k+\tau} (\zeta - t_k)^3 \cos d_i(t_k + \tau - \zeta) d\zeta,
\end{aligned} \tag{2.32}$$

where $\mathbf{V}_0, \mathbf{V}_1$ are diagonal matrices with diagonal elements $-d_i \sin d_i\tau, \cos d_i\tau$, respectively.

3. Energy Conservation Equation

One reason of expressing the dynamic equation in the form of (2.1) is to establish the energy conservation equation more conveniently and more directly. The following steps illustrate the construction of the energy conservation equation. Multiplying $\dot{\mathbf{x}}^T$ to both sides of (2.1), we have

$$\dot{\mathbf{x}}^T \mathbf{M} \ddot{\mathbf{x}} + \dot{\mathbf{x}}^T \mathbf{K} \mathbf{x} = \dot{\mathbf{x}}^T \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, t). \quad (3.1)$$

Integrating (2.32) from t_k to t_{k+1} , we can obtain

$$\int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}^T \mathbf{M} \ddot{\mathbf{x}} dt + \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}^T \mathbf{K} \mathbf{x} dt = \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}^T \mathbf{f} dt \quad k = 0, 1, 2, 3, \dots, \quad (3.2)$$

namely,

$$\frac{1}{2} \dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}} \Big|_{t_k}^{t_{k+1}} + \frac{1}{2} \dot{\mathbf{x}}^T \mathbf{K} \mathbf{x} \Big|_{t_k}^{t_{k+1}} = \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}^T \mathbf{f} dt \quad k = 0, 1, 2, 3, \dots \quad (3.3)$$

Simplifying (3.3), we can obtain an energy conservation equation between t_k and t_{k+1} as follows:

$$T_{k+1} - T_k + V_{k+1} - V_k = \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}^T \mathbf{f} dt \quad k = 0, 1, 2, 3, \dots, \quad (3.4)$$

where

$$T_{k+1} = \frac{1}{2} \dot{\mathbf{x}}_{k+1}^T \mathbf{M} \dot{\mathbf{x}}_{k+1}, \quad T_k = \frac{1}{2} \dot{\mathbf{x}}_k^T \mathbf{M} \dot{\mathbf{x}}_k, \quad V_{k+1} = \frac{1}{2} \dot{\mathbf{x}}_{k+1}^T \mathbf{K} \mathbf{x}_{k+1}, \quad V_k = \frac{1}{2} \dot{\mathbf{x}}_k^T \mathbf{K} \mathbf{x}_k. \quad (3.5)$$

Substituting (2.30) and (2.31) into the left hand side of (3.4), a polynomial of the undermined parameter β can be easily achieved. For the right hand side of (3.4), the integral term can be firstly decomposed into two parts as follows:

$$f_1 = \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}^T \mathbf{q}_1(\mathbf{x}) dt, \quad f_2 = \int_{t_k}^{t_{k+1}} \dot{\mathbf{x}}^T \mathbf{q}_2(\dot{\mathbf{x}}, t) dt, \quad (3.6)$$

where the term f_1 is an integral of an autonomous system and can be integrated easily. The $\dot{\mathbf{x}}(t)$ in term f_2 can be expressed as a polynomial of time using the relationship in (2.4) and taking the derivative of (2.15) with respect to time. Two predictive methods are recommended for determining the unknown term \mathbf{y}_1 in (2.15). One is to let $\mathbf{y}_1 = \mathbf{y}_0 + \dot{\mathbf{y}}_0 \boldsymbol{\tau}$ and the other is to let $\beta = 1$ in (2.23).

Then through (3.5), an algebraic equation with an undetermined parameter β can be established and β can be numerically obtained by the Newton iteration method or other algebraic methods. Finally, substituting β into (2.30) and (2.31), a numerical result can then be achieved.

4. Calculations of the Interpolation

Before giving some numerical examples, choosing the proper interpolation form of (2.18) must be discussed because it will affect the accuracy and stability of the proposed algorithm. In the current study, the authors use the Hermite interpolation to approximate the $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4$ in (2.18), that is,

$$\begin{Bmatrix} r_0^i \\ r_1^i \\ r_2^i \\ r_3^i \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{\tau} & 0 & \frac{1}{\tau} & 0 \\ -\frac{3}{\tau^2} & -\frac{2}{\tau} & \frac{3}{\tau^2} & -\frac{1}{\tau} \\ \frac{2}{\tau^3} & \frac{1}{\tau^2} & -\frac{2}{\tau^3} & \frac{1}{\tau^2} \end{bmatrix} \cdot \begin{Bmatrix} f_i(0, x_i(0), \dot{x}_i(0)) \\ f_i'(0, x_i(0), \dot{x}_i(0)) \\ f_i(t, x_i(\tau), \dot{x}_i(\tau)) \\ f_i'(t, x_i(\tau), \dot{x}_i(\tau)) \end{Bmatrix}. \quad (4.1)$$

It should be noted that there are unknowns in the right hand term of (4.1) which are $x_i(\tau), \dot{x}_i(\tau)$. The prediction of the two unknowns is shown below. For example, we can let

$$\begin{aligned} x_i(\tau) &= x_i(0) + \dot{x}_i(0) \cdot \tau, \\ \dot{x}_i(\tau) &= \dot{x}_i(0) + \ddot{x}_i(0) \cdot \tau. \end{aligned} \quad (4.2)$$

Then at every iteration of (3.4), the parameter β can be updated. Submitting β into (2.30) and (2.31), a new prediction of the displacement and velocity can therefore be obtained.

5. Numerical Examples

In this section we give some numerical examples to verify the effectiveness of the proposed algorithm, in particular, the advantage in stability of the proposed algorithm. Since (2.14) is a fourth order Taylor expansion, the energy conservation algorithm has fourth order accuracy. So we choose the Rounge-Kutta method as a numerical comparison. The numerical results show the advantages of the proposed energy conservation algorithm in terms of its integration stability and the ability to eliminate the algorithm damping inherent in the Rounge-Kutta method.

5.1. The Oscillation of a Nonlinear Simple Pendulum

The dynamic equation of a nonlinear single pendulum without damping can be written as

$$\ddot{x} + \omega_0^2 \sin x = 0, \quad \omega_0^2 = 1.0, \quad x(0) = 1.57, \quad (5.1)$$

where x denotes the angular displacement. The numerical solutions are shown in Figure 1. From the figure we can see that the proposed energy conservation method (ECM) can keep the numerical stability and have no computing damping under large-step comparing with the Rounge-Kutta (RK) method. The numerical result of parameter β is shown in Figure 2. Table 1

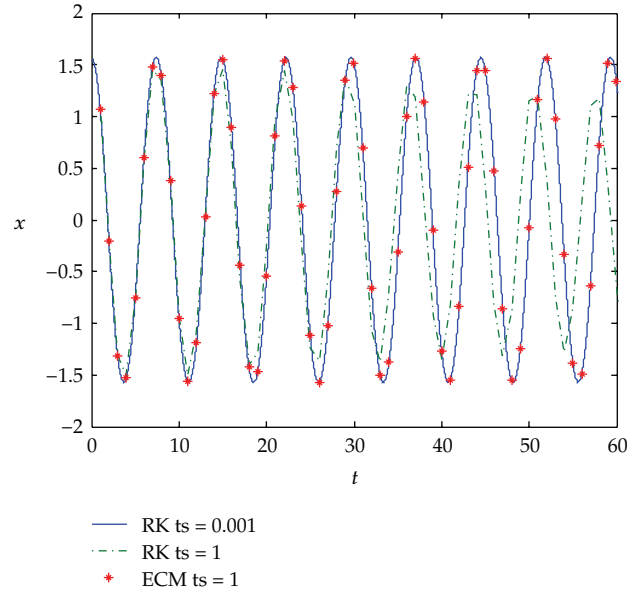


Figure 1: Angular Displacement comparison between the proposed energy conservation method (ECM) and the RK method.

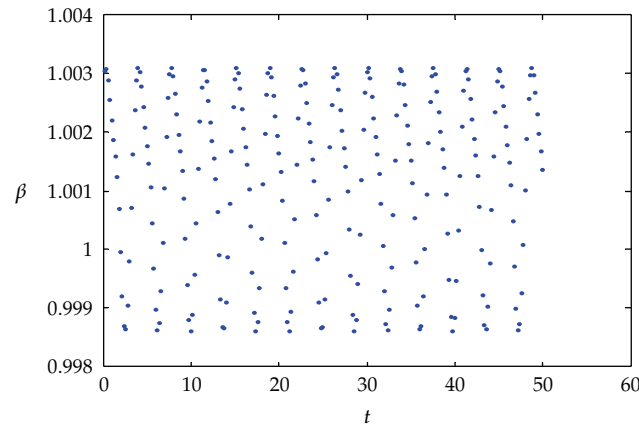


Figure 2: Value of parameter β (time step = 1.0 s).

gives the comparison of the computing efficiency. The efficiency of the proposed method is not as good as the RK method due to the iteration of parameter β and the time needed to compute the associated matrices. Figure 3 gives the error analysis between the ECM and the RK under time step 1.0 s.

5.2. The Unforced Linear Vibration of the Cuboid Rigid Body with Two DOF

The structural diagram of the system is shown in Figure 4. The mass of the rigid body is m and the length of the hemline is a . The center of mass is collocated at the geometry center

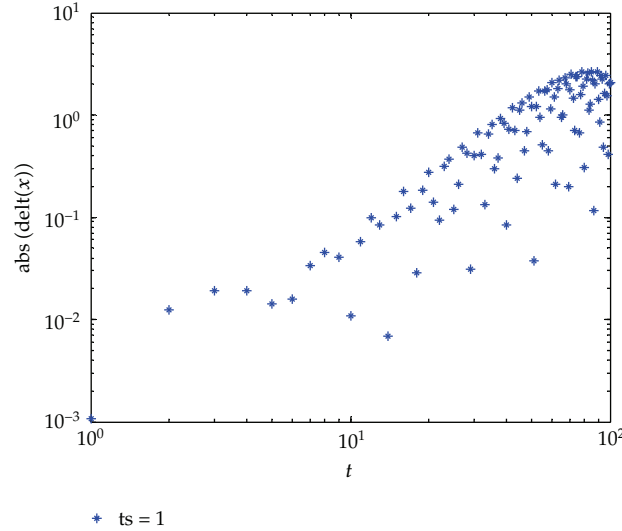


Figure 3: The log-log plot of the error between the ECM and RK.

Table 1: Comparison of computing efficiency.

	CPU	Memory	Integrations	Steps	End time	Time elapsed
RK	Intel core2 2.26G	2 G	500000	0.001 s	500 s	5.736 s
ECM	Intel core2 2.26G	2 G	500	1.0 s	500 s	1 m 7.123 s

(point C). The mass moment of inertia around the center of mass is J and the stiffness of the spring is k . The deformations of the two springs are x_1, x_2 . The displacement in vertical direction of the center of mass is x_c . The angular displacement of the rigid body about the mass center is ϕ . Using the above parameters, the equation of motion of the system can be written as

$$\begin{bmatrix} \frac{m}{4} + \frac{J}{a^2} & \frac{m}{4} - \frac{J}{a^2} \\ \frac{m}{4} - \frac{J}{a^2} & \frac{m}{4} + \frac{J}{a^2} \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}. \quad (5.2)$$

Let $m = 8$, $a = 1$, $k = 2$, $J = 1$, $x_1 = 1$, and $x_2 = -1$. Figures 5 and 6 compare the displacement (x_1) and velocity (x_2 -dot) results predicted by the proposed method and the RK method. It can be seen that even with a big time step 1.0 s, the proposed method still has an accurate numerical solution but the RK method does not. Table 2 gives the comparison of the computing efficiency. Again, the efficiency of the proposed method is lower than that of the RK method in calculating these two degrees of freedom problem. Furthermore, it is noticed that the RK method almost keeps the same efficiency in Sections 5.1 and 5.2.

Figure 7 gives the error analysis between the ECM and the RK under time step 1.0 s. As Figure 7 already shows that the accuracy of the ECM is almost same as the result of RK with a 0.001 time step, the comparison does not use the RK with a small time step.

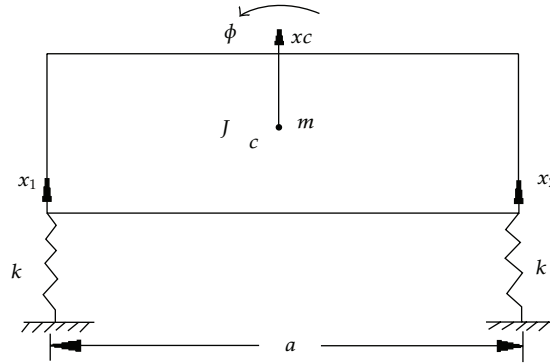


Figure 4: Structural diagram of the cuboid rigid body.

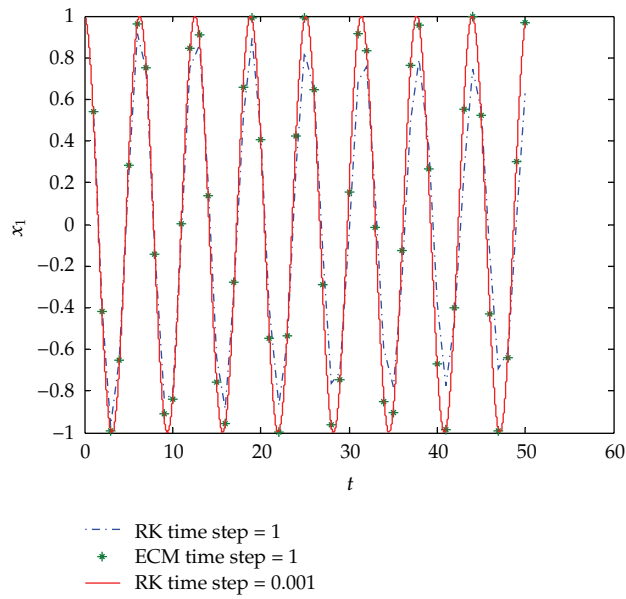


Figure 5: Comparison of displacement.

5.3. The Unforced Nonlinear Oscillation of a Spring Pendulum with Two DOF

The dynamic equation of the spring pendulum can be written as

$$\begin{aligned} \ddot{x}_1 + 2c_1\dot{x}_1 + \omega_1^2x_1 - b_1x_1x_2 &= 0, \\ \ddot{x}_2 + 2c_2\dot{x}_2 + \omega_2^2x_2 - b_2x_1^2 &= 0. \end{aligned} \tag{5.3}$$

Figures 8 and 9 show the numerical solution under different damping. Parameters and initial condition are given as follows:

$$\omega_1 = 1.0, \quad \omega_2 = 1.5, \quad b_1 = b_2 = 1.0, \quad x_1 = x_2 = 0.1, \quad \dot{x}_1 = \dot{x}_2 = 0.0. \tag{5.4}$$

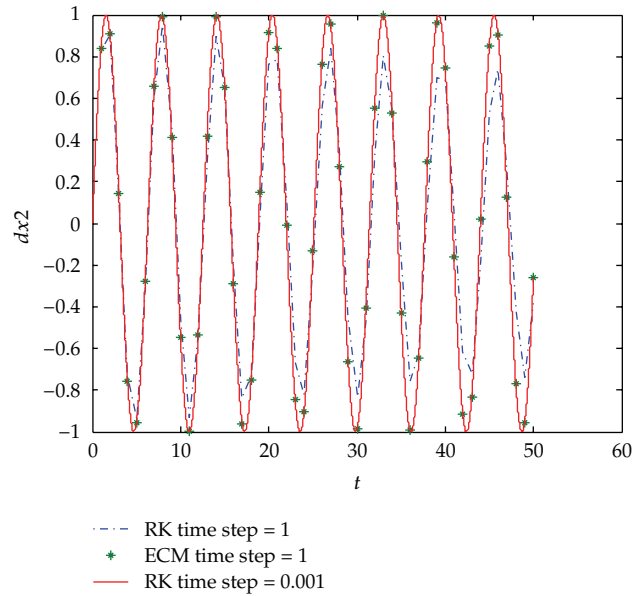


Figure 6: Comparison of velocity.

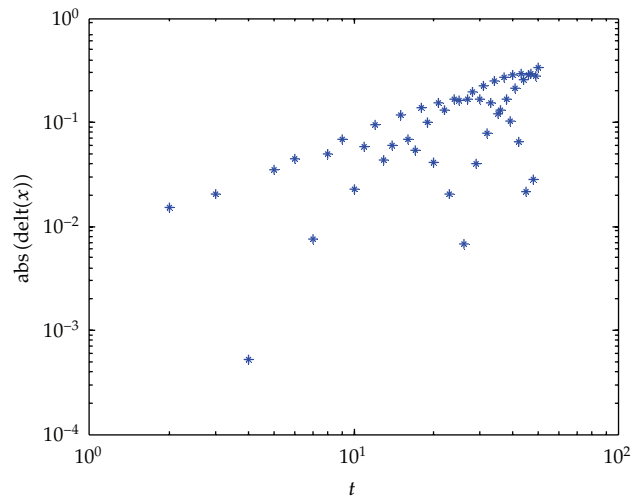


Figure 7: The log-log plot of the error between the ECM and RK with time step 1.0 s.

Figure 10 shows the comparison of the numerical results between ECM and RK methods under large time steps. It is obvious that the proposed method can eliminate algorithm damping better and provides better stability than the RK scheme. Parameters and initial conditions used in the calculation are as follows:

$$\begin{aligned}
 c_1 = c_2 = 0.0, \quad \omega_1 = 1.0, \quad \omega_2 = 1.5, \\
 b_1 = b_2 = 1.0, \quad x_1 = x_2 = 0.1, \quad \dot{x}_1 = \dot{x}_2 = 0.0.
 \end{aligned}
 \tag{5.5}$$

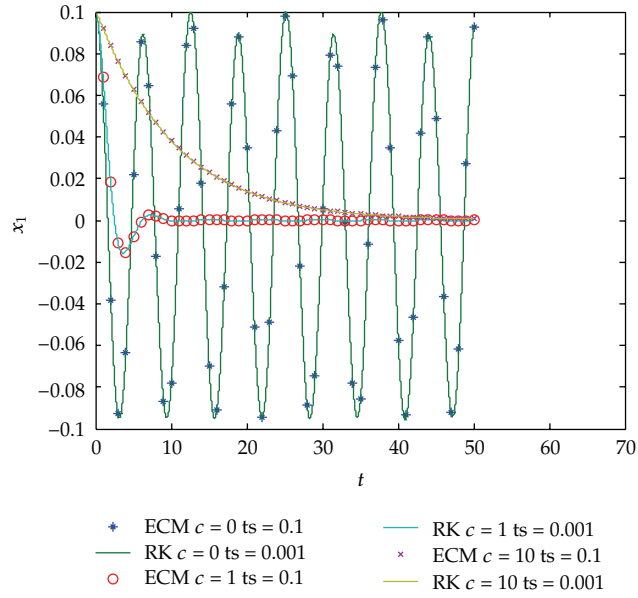


Figure 8: Displacement trajectory.

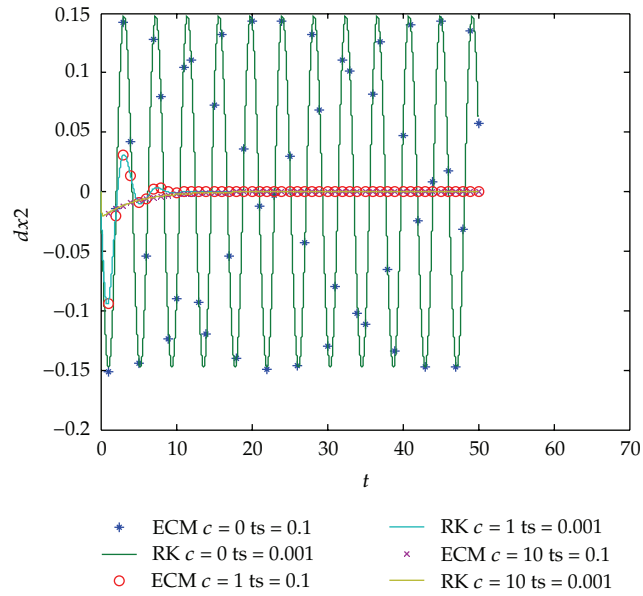


Figure 9: Velocity trajectory.

Table 2: Comparison of computing efficiency.

	CPU	Memory	Integrations	Steps	End time	Time elapsed
RK	Intel core2 2.26G	2 G	500000	0.001 s	500 s	5.849 s
ECM	Intel core2 2.26G	2 G	500	1.0 s	500 s	1 m 11.741 s

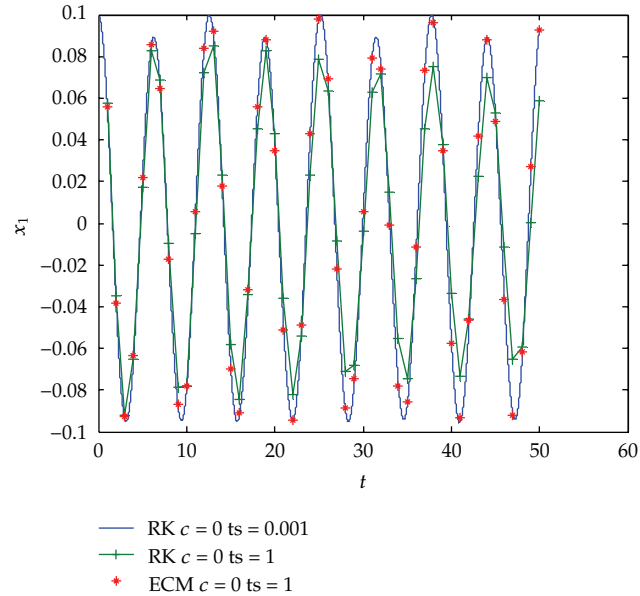


Figure 10: Displacement trajectory.

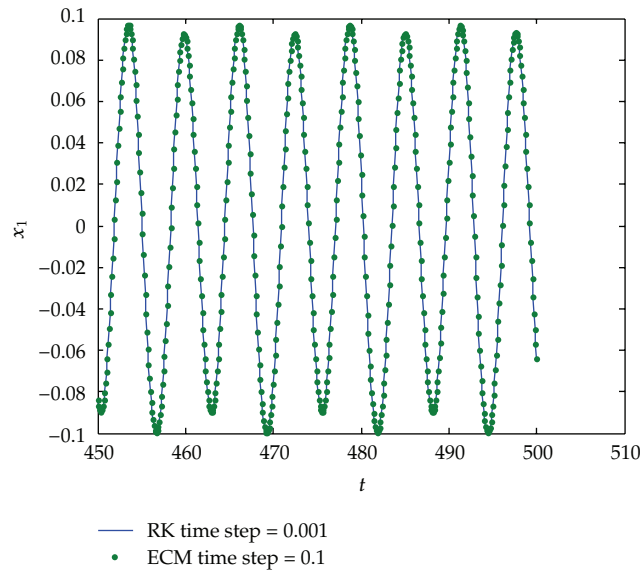


Figure 11: Displacement trajectory.

Figure 11 shows long-time response of (5.2). Parameter and initial condition is as same as (5.4). From this figure we can see that after long-term iteration the proposed method still keeps numerical stability under a relatively large time step. But the algorithm damping makes the RK lose accuracy. Table 3 gives the comparison of the computing efficiency between RK and the ECM algorithms. It is shown that, in solving the two degrees of freedom nonlinear problem, the efficiency of the EMC is lower than the RK method. Moreover, comparing the

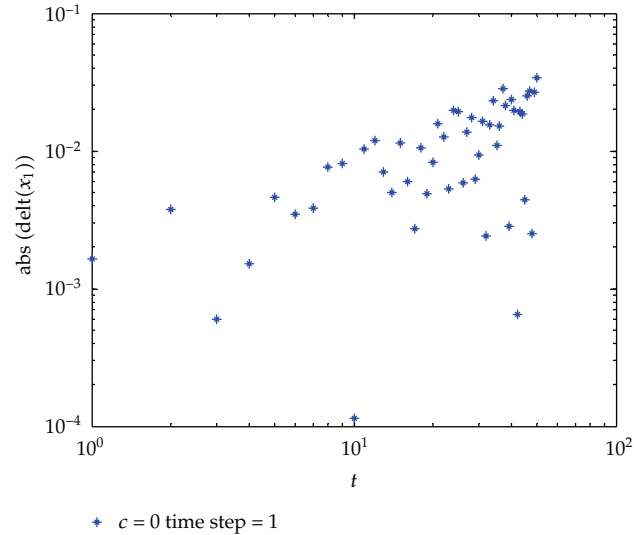


Figure 12: The log-log plot of the error between the ECM and RK.

Table 3: Comparison of computing efficiency.

	CPU	Memory	Integrations	Steps	End time	Time consuming	
RK	Intel core2	2.26G	2 G	500000	0.001 s	500 s	6.655 s
ECM	Intel core2	2.26G	2 G	500	1.0 s	500 s	4 m 18.263 s

elapsed time by the ECM in Table 2 and Table 3, it also can be seen that the computation efficiency of the ECM is worse for calculating nonlinear problems than for linear problems. Figure 12 shows the error analysis between the ECM and the RK under time step 1.0 s.

6. Conclusion

(1) The energy conservation algorithm has the advantage in stability and time step compared with some numerical means because the numerical solution has been corrected by the energy conservation equation.

(2) All examples have shown that the energy conservation method can eliminate algorithm damping. It is also an effective means for calculating the long-term characteristics of nonlinear dynamic systems.

(3) The proposed method conserves the angular momentum automatically. Although the efficiency of the energy conservation method is not as good as the RK algorithm as well as some other numerical methods discussed in the literature, the integration step is large enough to implement long-term integration with good numerical stability.

(4) The reason of the low efficiency of the proposed method is because the iterations need to calculate the parameter β and the time consumed in matrix computing needed by the algorithm. The efficiency of the EMC is lower in dealing with nonlinear problems compared with linear problems.

Acknowledgments

This work was funded by the “973” National Basic Research Project of China (no. Q10110919), Key Project of the National Natural Science Foundation of China (no. 10932003), “863” Project of China (no. 2009AA04Z101), and “973” National Basic Research Project of China (no. 2010CB832700). These supports are gratefully acknowledged. Many thanks are due to the reviewers for their valuable comments.

References

- [1] J. Chung and G. M. Hulbert, “A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method,” *Journal of Applied Mechanics*, vol. 60, no. 2, pp. 371–375, 1993.
- [2] H. M. Hilber, T. J. R. Huges, and R. L. Taylor, “Improved numerical dissipation for time integration algorithms in structural dynamics,” *Earthquake Engineering and Structural Dynamics*, vol. 5, pp. 283–292, 1977.
- [3] W. L. Wood, M. Bossak, and O. C. Zienkiewicz, “An alpha modification of Newmark’s method,” *International Journal for Numerical Methods in Engineering*, vol. 15, pp. 1562–1566, 1981.
- [4] T. C. Fung, “Unconditionally stable time-step-integration algorithms based on Hamilton’s principle,” *AIAA Journal*, vol. 38, no. 8, pp. 1443–1464, 2000.
- [5] T. C. Fung, “On the equivalence of the time domain differential quadrature method and the dissipative Runge-Kutta collocation method,” *International Journal for Numerical Methods in Engineering*, vol. 53, no. 2, pp. 409–431, 2002.
- [6] T. C. Fung, “Construction of higher-order accurate time-step integration algorithms by equal-order polynomial projection,” *Journal of Vibration and Control*, vol. 11, no. 1, pp. 19–49, 2005.
- [7] T. C. Fung, “Bi-discontinuous time step integration algorithms-Part 2: second-order equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 3–4, pp. 351–374, 2003.
- [8] T. C. Fung and Z. L. Chen, “Krylov precise time-step integration method,” *International Journal for Numerical Methods in Engineering*, vol. 68, no. 11, pp. 1115–1136, 2006.
- [9] J. C. Simo and O. Gonzalez, “Assessment of energy-momentum and symplectic schemes for stiff dynamic systems,” in *Proceedings of the ASME Winter Annual Meeting*, New Orleans, La, USA, December 1993.
- [10] Q. V. Bui, “Modified Newmark family for non-linear dynamic analysis,” *International Journal for Numerical Methods in Engineering*, vol. 61, no. 9, pp. 1390–1420, 2004.
- [11] R. A. LaBudde and D. Greenspan, “Energy and momentum conserving methods of arbitrary order for the numerical integration of equations of motion, II. Motion of a system of particles,” *Numerische Mathematik*, vol. 26, no. 1, pp. 1–16, 1976.
- [12] T. J. R. Hughes, T. K. Caughey, and W. K. Liu, “Finite-element methods for nonlinear elastodynamics which conserve energy,” *Journal of Applied Mechanics*, vol. 45, no. 2, pp. 366–370, 1978.
- [13] D. Greenspan, “Conservative numerical methods for $\ddot{x} = f(x)$,” *Journal of Computational Physics*, vol. 56, no. 1, pp. 28–41, 1984.
- [14] J. C. Simo and N. Tarnow, “The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics,” *Journal of Applied Mathematics and Physics (ZAMP)*, vol. 43, no. 5, pp. 757–792, 1992.
- [15] J. C. Simo, N. Tarnow, and K. K. Wong, “Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 100, no. 1, pp. 63–116, 1992.
- [16] D. Greenspan, “Completely conservative, covariant numerical methodology,” *Computers and Mathematics with Applications*, vol. 29, no. 4, pp. 37–43, 1995.
- [17] T. C. Fung and S. K. Chow, “Solving non-linear problems by complex time step methods,” *Communications in Numerical Methods in Engineering*, vol. 18, no. 4, pp. 287–303, 2002.
- [18] U. M. Ascher and S. Reich, “The midpoint scheme and variants for Hamiltonian systems: advantages and pitfalls,” *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 1045–1065, 1999.
- [19] U. M. Ascher and S. Reich, “On some difficulties in integrating highly oscillatory Hamiltonian systems,” in *Proceedings on Computational Molecular Dynamics*, pp. 281–296, Springer Lecture Notes, 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

