Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Computational Physics 214 (2006) 505–520
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

High-order predictor–corrector of exponential fitting for the *N*-body problems

Chen Tang [a,*], Wenping Wang [a], Haiqing Yan [a], Zhanqing Chen [b]

[a] *Department of Applied Physics, University of Tianjin, Tianjin 300072, PR China*
[b] *Department of Mechanical Engineering and Automation, Harbin Institute of Technology, PR China*

Received 11 April 2005; received in revised form 13 September 2005; accepted 28 September 2005
Available online 11 November 2005

Abstract

We develop the seventh-step predictor–corrector of exponential fitting method for the *N*-body problems. We apply our proposed scheme to Kepler problem, the interaction of seven argon atoms in a plane and three particles bound together by two springs of different stiffnesses. The three problems have various potential functions. We demonstrate the accuracy and efficiency of our proposed scheme via comparison with other analytical and numerical results. The numerical results show that the schemes are highly accurate and computationally efficient.

Keywords: *N*-body problems; Exponential time differencing; Exponential fitting; High-order schemes

1. Introduction

The *N*-body problem occurs in almost all branches of physics from studies on submicroscopic systems to macroscopic ones (Bose–Einstein condensation, molecular oscillations, protein-folding, granular dynamics, swarming, multilane traffic flow, galaxy formation, etc.). The problem is characterized by a set of coupled differential (or difference) equations. *N*-body problems are difficult to solve. In fact, 2-body problem has a well-known analytic solution [1], but *N*-body system is in general non-integrable [2]. The simulation of *N*-body problems typically involves two classes of physical phenomena: those governed by long-range forces and those governed by short-range forces [3]. Perhaps the best examples of these two extremes are gravity and gas dynamics. In the gravitational problem, every particle affects every other particle. In a gas dynamics simulation, each particle has a limited domain of influence which is much smaller than the entire computational domain. These classes of simulations have different computational goals in the creation of *N*-body algorithms. For gravitational forces, the computational goal is to reduce the work as much as possible with a controllable amount of error. This task is a *long-range* search. For simulations of gas dynamics, the computational goal is to identify as rapidly as possible all particles within a specified distance from the test particle. This task is a

* Corresponding author. Tel.: +86 022 87894268.
E-mail address: tangchen@tju.edu.cn (C. Tang).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.09.028

short-range search. Despite the fact that the long- and short-range problems represent different types of physical phenomena, there are many circumstances in which both gas dynamics and gravity are needed in the same simulation. To approximately solve N-body problems, one often attempts to discretize the equations of motion and study the evolution of the system numerically. However, discretization of a system of differential equations typically leads to a loss of accuracy. This often necessitates the use of small time steps, so that many iterations will be required.

In this paper, we develop high-order predictor–corrector of exponential fitting method for the N-body problems. High-order methods have the advantage of smaller error constant in the truncation error. In the last decade, the numerical schemes of exponential time differencing'' (ETD) scheme [4,5] or called ''exponential fitting''(EF) [6] have been constructed in different ways. Explicit multistep exponential fitting with various order accuracy were constructed [7,8]. Runge–Kutta EF schemes were derived [8–10]. Optimal implicit exponentially-fitted Runge–Kutta methods and implicit ETD schemes of arbitrary order were developed [11,12]. In addition, exponentially fitted variable two-step BDF algorithm for first order ODEs was constructed [13]. We have developed the various order explicit and implicit multistep schemes of exponential fitting for systems of ordinary differential equations [14,15]. Here, we apply high-order predictor–corrector of exponential time differencing to Kepler problem, the interaction of seven argon atoms in a plane and three particles bound together by two springs of different stiffnesses, which have various potential functions, and compare with other analytical and numerical results. Although the scheme is neither reversible nor symplectic, the numerical results show that the schemes are highly accurate and computationally efficient.

2. High-order predictor–corrector of exponential fitting (PCEF)

2.1. The derivation of high-order PCEF

We have derived explicit and implicit exponential time differencing schemes of arbitrary order [14,15]. Here, we provide the seventh-step explicit and implicit exponential time differencing schemes.

The nth-order system of first-order differential equations for initial value problems in form

$$\begin{cases} \dot{y}_1 = f_1(t, y_1, \ldots, y_n), \\ \vdots \\ \dot{y}_j = f_j(t, y_1, \ldots, y_n), \\ \vdots \\ \dot{y}_n = f_n(t, y_1, \ldots, y_n). \end{cases} \tag{1}$$

We introduce the vectors

$$\mathbf{y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}, \quad \mathbf{f} = (f_1, f_2, \ldots, f_n)^{\mathrm{T}}.$$

Eq. (1) is expressed as

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t). \tag{2}$$

Introducing a constant matrix $\mathbf{H}$, we rearrange Eq. (2) as

$$\dot{\mathbf{y}} - \mathbf{H}\mathbf{y} = \mathbf{F}, \tag{3}$$

where

$$\mathbf{F} = \mathbf{f}(t, \mathbf{y}) - \mathbf{H}\mathbf{y} \tag{4}$$

and the constant matrix $\mathbf{H}$ should satisfy $\det(\mathbf{H}) \neq 0$.

We multiply (3) through by an integrating factor $\exp(-\mathbf{H}t)$, then integrate from t_k to t_{k+1} to give

$$\mathbf{y}_{k+1} = \exp(\mathbf{H}h) \cdot \mathbf{y}_k + \int_{t_k}^{t_{k+1}} \exp \mathbf{H}(t_{k+1} - t) \cdot \mathbf{F}(t)\, \mathrm{d}t. \tag{5}$$

Let $t_{k+1} = t_k + h$, Eq. (5) is expressed as

$$\mathbf{y}_{k+1} = \exp(\mathbf{H}h)\cdot\mathbf{y}_k + \int_{t_k}^{t_k+h} \exp\mathbf{H}(t_k + h - t)\cdot\mathbf{F}(t)\,\mathrm{d}t. \tag{6}$$

Let $t = t_k + \tau$, we have

$$\mathbf{y}_{k+1} = \exp(\mathbf{H}h)\cdot\mathbf{y}_k + \int_{0}^{h} \exp\mathbf{H}(h - \tau)\cdot\mathbf{F}(t_k + \tau)\,\mathrm{d}\tau. \tag{7}$$

Since the approximations $\mathbf{y}_{k-6},\ldots,\mathbf{y}_{k-1},\mathbf{y}_k$ at time points $t_{k-6},\ldots,t_{k-1},t_k$ are known, the values $\mathbf{F}_{k-6},\ldots,\mathbf{F}_{k-1},\mathbf{F}_k$ at time points $t_{k-6},\ldots,t_{k-1},t_k$ are also available by the formula (4). It is natural to replace the function $\mathbf{F}(t_k+\tau)$ in (7) by interpolating polynomial $\mathbf{N}(t_k+\tau)$ through the points $(t_k,\mathbf{F}_k),(t_{k-1},\mathbf{F}_{k-1}),\ldots,(t_{k-6},\mathbf{F}_{k-6})$. The Newton interpolating polynomial formed through $(t_k,F_k),(t_{k-1},\mathbf{F}_{k-1}),\ldots,(t_{k-6},\mathbf{F}_{k-6})$ can be expressed as

$$\mathbf{F}(t_k+\tau) \approx \mathbf{N}(t_k+\tau) = \sum_{s=0}^{6}(-1)^s \sum_{j=s}^{6}(-1)^j \binom{\frac{-\tau}{h}}{j}\binom{j}{s}\mathbf{F}_{k-s}, \tag{8}$$

where

$$\binom{\frac{-\tau}{h}}{j} = \frac{\left(\frac{-\tau}{h}\right)\left(\frac{-\tau}{h}-1\right)\cdots\left(\frac{-\tau}{h}-j+1\right)}{j!}, \quad \text{and} \quad \binom{\frac{-\tau}{h}}{0} = 1. \tag{9}$$

Inserting (8) into (7), we obtain the seventh-step explicit EF scheme

$$\mathbf{y}_{k+1} = \exp(\mathbf{H}h)\cdot\mathbf{y}_k + h\left[(-1)^0\sum_{j=0}^{6}\mathbf{g}_j\binom{j}{0}\right]\mathbf{F}_k + h\left[(-1)^1\sum_{j=1}^{6}\mathbf{g}_j\binom{j}{1}\right]\mathbf{F}_{k-1} + \cdots + h\mathbf{g}_6\mathbf{F}_{k-6}, \tag{10}$$

where

$$\mathbf{g}_j = \int_0^1 \exp\mathbf{H}h(1-\tau)\cdot(-1)^j\binom{-\tau}{j}\,\mathrm{d}\tau. \tag{11}$$

Considering Eq. (9), the seventh-step explicit EF scheme is consequently expressed as

$$\begin{aligned}\mathbf{y}_{k+1} = {} & \exp(\mathbf{H}h)\cdot\mathbf{y}_k + h(\mathbf{g}_0+\mathbf{g}_1+\mathbf{g}_2+\mathbf{g}_3+\mathbf{g}_4+\mathbf{g}_5+\mathbf{g}_6)\mathbf{F}_k - h(\mathbf{g}_1+2\mathbf{g}_2+3\mathbf{g}_3+4\mathbf{g}_4+5\mathbf{g}_5+6\mathbf{g}_6)\mathbf{F}_{k-1}\\ & + h(\mathbf{g}_2+3\mathbf{g}_3+6\mathbf{g}_4+10\mathbf{g}_5+15\mathbf{g}_6)\mathbf{F}_{k-2} - h(\mathbf{g}_3+4\mathbf{g}_4+10\mathbf{g}_5+20\mathbf{g}_6)\mathbf{F}_{k-3} + h(\mathbf{g}_4+5\mathbf{g}_5+15\mathbf{g}_6)\mathbf{F}_{k-4}\\ & - h(\mathbf{g}_5+6\mathbf{g}_6)\mathbf{F}_{k-5} + h\mathbf{g}_6\mathbf{F}_{k-6}.\end{aligned} \tag{12}$$

The coefficients $\mathbf{g}_0,\mathbf{g}_1,\mathbf{g}_2,\mathbf{g}_3,\mathbf{g}_4,\mathbf{g}_5,\mathbf{g}_6$ can be derived by Eq. (11).

$$\mathbf{g}_0 = -\frac{\mathbf{H}^{-1}}{h}(\mathbf{I}-\mathbf{T}), \tag{13-a}$$

$$\mathbf{g}_1 = -\frac{\mathbf{H}^{-1}}{h}\mathbf{I} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^2(\mathbf{I}-\mathbf{T}), \tag{13-b}$$

$$\mathbf{g}_2 = -\frac{\mathbf{H}^{-1}}{h}\mathbf{I} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{3}{2}\mathbf{I}-\frac{1}{2}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3(\mathbf{I}-\mathbf{T}), \tag{13-c}$$

$$\mathbf{g}_3 = -\frac{\mathbf{H}^{-1}}{h}\mathbf{I} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{11}{6}\mathbf{I}-\frac{1}{3}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3(2\mathbf{I}-\mathbf{T}) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4(\mathbf{I}-\mathbf{T}), \tag{13-d}$$

$$\begin{aligned}\mathbf{g}_4 = {} & -\frac{\mathbf{H}^{-1}}{h}\mathbf{I} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{25}{12}\mathbf{I}-\frac{1}{4}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3\left(\frac{35}{12}\mathbf{I}-\frac{11}{12}\mathbf{T}\right)\\ & - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4\left(\frac{5}{2}\mathbf{I}-\frac{3}{2}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^5(\mathbf{I}-\mathbf{T}),\end{aligned} \tag{13-e}$$

$$\mathbf{g}_5 = -\frac{\mathbf{H}^{-1}}{h}\mathbf{I} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{137}{60}\mathbf{I} - \frac{1}{5}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3\left(\frac{15}{4}\mathbf{I} - \frac{5}{6}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4\left(\frac{17}{4}\mathbf{I} - \frac{7}{4}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^5(3\mathbf{I} - 2\mathbf{T}) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^6(\mathbf{I} - \mathbf{T}), \tag{13-f}$$

$$\mathbf{g}_6 = -\frac{\mathbf{H}^{-1}}{h}\mathbf{I} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{49}{20}\mathbf{I} - \frac{1}{6}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3\left(\frac{203}{45}\mathbf{I} - \frac{137}{180}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4\left(\frac{49}{8}\mathbf{I} - \frac{15}{8}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^5\left(\frac{35}{6}\mathbf{I} - \frac{17}{6}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^6\left(\frac{7}{2}\mathbf{I} - \frac{5}{2}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^7(\mathbf{I} - \mathbf{T}), \tag{13-g}$$

where $\mathbf{T} = \exp(\mathbf{H}h)$.

The seventh-step explicit EF scheme is obtained by integrating the interpolation polynomial formed through the points $(t_k, \mathbf{F}_k), (t_{k-1}, \mathbf{F}_{k-1}), \ldots, (t_{k-6}, \mathbf{F}_{k-6})$ form t_k to t_{k+1}. The seventh-step implicit EF scheme is also obtained by the interpolation polynomial which uses in addition the $(t_{k+1}, \mathbf{F}_{k+1})$.

Let $t_k = t_{k+1} - h$, Eq. (5) is expressed as

$$\mathbf{y}_{k+1} = \exp(\mathbf{H}h)\cdot\mathbf{y}_k + \int_{t_{k+1}-h}^{t_{k+1}} \exp\mathbf{H}(t_{k+1} - t)\cdot\mathbf{F}(t)\,\mathrm{d}t. \tag{14}$$

Let $t = t_{k+1} - \tau$, we have

$$\mathbf{y}_{k+1} = \exp(\mathbf{H}h)\cdot\mathbf{y}_k + \int_0^h \exp\mathbf{H}\tau\cdot\mathbf{F}(t_{k+1} - \tau)\,\mathrm{d}\tau. \tag{15}$$

The Newton interpolation polynomial formed through $(t_{k+1}, \mathbf{F}_{k+1}), (t_k, \mathbf{F}_k), \ldots, (t_{k-5}, \mathbf{F}_{k-5})$ can be expressed as

$$\mathbf{F}(t_{k+1} - \tau) \approx \mathbf{N}(t_{k+1} - \tau) = \sum_{s=0}^{6}(-1)^s\sum_{j=s}^{6}(-1)^j\binom{\frac{\tau}{h}}{j}\binom{j}{s}\mathbf{F}_{k-s+1}. \tag{16}$$

Inserting (16) into (15), we obtain the seventh-step implicit EF scheme

$$\mathbf{y}_{k+1} = \exp(\mathbf{H}h)\cdot\mathbf{y}_k + h\left[(-1)^0\sum_{j=0}^{6}\mathbf{g}_j^*\binom{j}{0}\right]\mathbf{F}_{k+1} + h\left[(-1)^1\sum_{j=1}^{6}\mathbf{g}_j^*\binom{j}{1}\right]\mathbf{F}_k + \cdots + h\mathbf{g}_6^*\mathbf{F}_{k-5}, \tag{17}$$

where the coefficients $\mathbf{g}_j^*$ satisfy

$$\mathbf{g}_j^* = \int_0^1 \exp\mathbf{H}h\tau\cdot(-1)^j\binom{\tau}{j}\,\mathrm{d}\tau = \int_{-1}^0 \exp\mathbf{H}h(-\tau)\cdot(-1)^j\binom{-\tau}{j}\,\mathrm{d}\tau. \tag{18}$$

Considering Eq. (9) the seventh-step implicit EF scheme is consequently expressed as

$$\begin{aligned}\mathbf{y}_{k+1} = {} & \exp(\mathbf{H}h)\cdot\mathbf{y}_k + h(\mathbf{g}_0^* + \mathbf{g}_1^* + \mathbf{g}_2^* + \mathbf{g}_3^* + \mathbf{g}_4^* + \mathbf{g}_5^* + \mathbf{g}_6^*)\mathbf{F}_{k+1} - h(\mathbf{g}_1^* + 2\mathbf{g}_2^* + 3\mathbf{g}_3^* + 4\mathbf{g}_4^* + 5\mathbf{g}_5^* + 6\mathbf{g}_6^*)\mathbf{F}_k \\ & + h(\mathbf{g}_2^* + 3\mathbf{g}_3^* + 6\mathbf{g}_4^* + 10\mathbf{g}_5^* + 15\mathbf{g}_6^*)\mathbf{F}_{k-1} - h(\mathbf{g}_3^* + 4\mathbf{g}_4^* + 10\mathbf{g}_5^* + 20\mathbf{g}_6^*)\mathbf{F}_{k-2} + h(\mathbf{g}_4^* + 5\mathbf{g}_5^* + 15\mathbf{g}_6^*)\mathbf{F}_{k-3} \\ & - h(\mathbf{g}_5^* + 6\mathbf{g}_6^*)\mathbf{F}_{k-4} + h\mathbf{g}_6^*\mathbf{F}_{k-5}.\end{aligned} \tag{19}$$

The coefficients $\mathbf{g}_0^*, \mathbf{g}_1^*, \mathbf{g}_2^*, \mathbf{g}_3^*, \mathbf{g}_4^*, \mathbf{g}_5^*, \mathbf{g}_6^*$ can be derived by Eq. (18).

$$\mathbf{g}_0^* = -\frac{\mathbf{H}^{-1}}{h}(\mathbf{I} - \mathbf{T}), \tag{20-a}$$

$$\mathbf{g}_1^* = -\frac{\mathbf{H}^{-1}}{h}\mathbf{T} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^2(\mathbf{I} - \mathbf{T}), \tag{20-b}$$

$$\mathbf{g}_2^* = -\left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3(\mathbf{I} - \mathbf{T}), \tag{20-c}$$

$$\mathbf{g}_3^* = -\left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{1}{3}\mathbf{I}+\frac{1}{6}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3\mathbf{I} - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4(\mathbf{I}-\mathbf{T}), \tag{20-d}$$

$$\mathbf{g}_4^* = -\left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{1}{4}\mathbf{I}+\frac{1}{12}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3\left(\frac{11}{12}\mathbf{I}+\frac{1}{12}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4\left(\frac{3}{2}\mathbf{I}-\frac{1}{2}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^5(\mathbf{I}-\mathbf{T}), \tag{20-e}$$

$$\mathbf{g}_5^* = -\left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{1}{5}\mathbf{I}+\frac{1}{20}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3\left(\frac{5}{6}\mathbf{I}+\frac{1}{12}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4\left(\frac{7}{4}\mathbf{I}-\frac{1}{4}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^5(2\mathbf{I}-\mathbf{T}) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^6(\mathbf{I}-\mathbf{T}), \tag{20-f}$$

$$\mathbf{g}_6^* = -\left(\frac{\mathbf{H}^{-1}}{h}\right)^2\left(\frac{1}{6}\mathbf{I}+\frac{1}{30}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^3\left(\frac{137}{180}\mathbf{I}+\frac{13}{180}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^4\left(\frac{15}{8}\mathbf{I}-\frac{1}{8}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^5\left(\frac{17}{6}\mathbf{I}-\frac{5}{6}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^6\left(\frac{5}{2}\mathbf{I}-\frac{3}{2}\mathbf{T}\right) - \left(\frac{\mathbf{H}^{-1}}{h}\right)^7(\mathbf{I}-\mathbf{T}). \tag{20-g}$$

Here the arbitrary-precision arithmetic is used to calculated both $\mathbf{g}_j$ and $\mathbf{g}_j^*$. We use the seventh-step explicit multistep EF method to obtain the predicted solution value, then use these predicted values in the corresponding implicit method to obtain the corrected solution value. This combination is called the seventh-step predictor–corrector EF method (PCEF7).

A multistep method normally needs the starting values. The starting values in the present multistep method are evaluated by the second-order Runge–Kutta EF scheme (RKEF2).

In the foregoing, we have derived the formulae in matrix form. It is inconvenient to implement the matrix calculation if the number of Eq. (1) n is very large. At this time, the matrix form cannot be adopted, and Eq. (1) is transformed equally to the following form

$$\begin{cases} \dot{y}_1 - c_1 y_1 = F_1(t, y_1, \ldots, y_n), \\ \vdots \\ \dot{y}_m - c_m y_m = F_m(t, y_1, \ldots, y_n), \\ \vdots \\ \dot{y}_n - c_n y_n = F_n(t, y_1, \ldots, y_n), \end{cases} \tag{21}$$

where $F_m(t, y_1, \ldots, y_n) = f_m(t, y_1, \ldots, y_n) - c_m y_m$, $m = 1, 2, \ldots, n$.

In this case, the seventh-step explicit EF scheme is expressed as

$$\begin{aligned} y_{m,k+1} = {} & \exp(c_m h)\cdot y_{m,k} + h(g_{m,0} + g_{m,1} + g_{m,2} + g_{m,3} + g_{m,4} + g_{m,5} + g_{m,6})F_{m,k} \\ & - h(g_{m,1} + 2g_{m,2} + 3g_{m,3} + 4g_{m,4} + 5g_{m,5} + 6g_{m,6})F_{m,k-1} \\ & + h(g_{m,2} + 3g_{m,3} + 6g_{m,4} + 10g_{m,5} + 15g_{m,6})F_{m,k-2} - h(g_{m,3} + 4g_{m,4} + 10g_{m,5} + 20g_{m,6})F_{m,k-3} \\ & + h(g_{m,4} + 5g_{m,5} + 15g_{m,6})F_{m,k-4} - h(g_{m,5} + 6g_{m,6})F_{m,k-5} + hg_{m,6}F_{m,k-6}. \end{aligned} \tag{22}$$

The seventh-step implicit EF scheme is expressed as

$$\begin{aligned} y_{m,k+1} = {} & \exp(c_m)\cdot y_{m,k} + h(g^*_{m,0} + g^*_{m,1} + g^*_{m,2} + g^*_{m,3} + g^*_{m,4} + g^*_{m,5} + g^*_{m,6})F_{m,k+1} \\ & - h(g^*_{m,1} + 2g^*_{m,2} + 3g^*_{m,3} + 4g^*_{m,4} + 5g^*_{m,5} + 6g^*_{m,6})F_{m,k} \\ & + h(g^*_{m,2} + 3g^*_{m,3} + 6g^*_{m,4} + 10g^*_{m,5} + 15g^*_{m,6})F_{m,k-1} - h(g^*_{m,3} + 4g^*_{m,4} + 10g^*_{m,5} + 20g^*_{m,6})F_{m,k-2} \\ & + h(g^*_{m,4} + 5g^*_{m,5} + 15g^*_{m,6})F_{m,k-3} - h(g^*_{m,5} + 6g^*_{m,6})F_{m,k-4} + hg^*_6 F_{m,k-5}. \end{aligned} \tag{23}$$

The coefficients $g_{m,0}, g_{m,1}, g_{m,2}, g_{m,3}, g_{m,4}, g_{m,5}, g_{m,6}$, $g^*_{m,0}, g^*_{m,1}, g^*_{m,2}, g^*_{m,3}, g^*_{m,4}, g^*_{m,5}, g^*_{m,6}$ have similar expression as Eqs. (13-a)–(13-f) and Eqs. (20-a)–(20-f). For example,

$$g_{m,1} = -\frac{c_m^{-1}}{h} - \left(\frac{c_m^{-1}}{h}\right)^2 (1 - \exp(c_m h)). \tag{24}$$

One may choose the matrix form PCEF7, also choose the non-matrix form PCEF7. If the matrix form is adopted, a given system is transformed equally to Eq. (3) form. If not, the system is transformed equally to Eq. (21).

For example, we consider Hamiltonians

$$H(p_i, q_i) = \frac{1}{2} p_i^T M^{-1} P_i + f(q_i), \tag{25}$$

where M is the $N \times N$ positive-definite mass matrix and f: $R^N \to R$ is the potential energy.

The Hamiltonians becomes

$$\begin{cases} \dot{q}_i = M^{-1} p_i \\ \dot{p}_i = -\nabla_{q_i} f(q_i) \end{cases} \quad i = 1, \ldots, N. \tag{26}$$

When the matrix form PCEF7 is adopted, Eq. (26) is transformed equally to the following form

$$\begin{bmatrix} \dot{q}_i \\ \dot{p}_i \end{bmatrix} - \begin{bmatrix} 0 & M^{-1} \\ I & 0 \end{bmatrix} \begin{bmatrix} q_i \\ p_i \end{bmatrix} = \begin{bmatrix} 0 \\ -q_i - \nabla_{q_i} f(q_i) \end{bmatrix}, \tag{27}$$

where I is the $N \times N$ identity matrix.

For this system,

$\mathbf{H}$ is $2N \times 2N$ constant matrix, i.e.,

$$\mathbf{H} = \begin{bmatrix} 0 & M^{-1} \\ I & 0 \end{bmatrix} \cdot \mathbf{F} = \begin{bmatrix} 0 \\ -q_i - \nabla_{q_i} f(q_i) \end{bmatrix}.$$

It is seen that the matrix H is a constant matrix, which should satisfy $\det(\mathbf{H}) \neq 0$ and should not be restricted to be diagonal. For a given system, the choice for the matrix H is easy. The matrix H depicts linear part of the system.

When the matrix form is not adopted, Eq. (26) is transformed equally to the following form

$$\begin{cases} \dot{q}_i - q_i = -q_i - M^{-1} p_i, \\ \dot{p}_i - p_i = -p_i - \nabla_{q_i} f(q_i). \end{cases} \tag{28}$$

The above derived formulae for coefficients $\mathbf{g}_j$, $\mathbf{g}_j^*$ seem to be complex. In fact, the present method is computationally very efficient. The coefficients $\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4, \mathbf{g}_5, \mathbf{g}_6$ and $\mathbf{g}_0^*, \mathbf{g}_1^*, \mathbf{g}_2^*, \mathbf{g}_3^*, \mathbf{g}_4^*, \mathbf{g}_5^*, \mathbf{g}_6^*$ are constant matrix and independent of the right functions $\mathbf{f}(\mathbf{y}, t)$ and $\mathbf{F}(\mathbf{y}, t)$, and can be easily evaluated in advance. Like Adam–Bashforth–Moulton predictor–corrector method, our EF predictor–corrector method requires only new force evaluation per integration step. All the other values of force have been calculated for earlier approximations.

2.2. Implementation of PCEF7

We summarize the steps to implement the matrix form PCEF7 scheme for Eq. (3). A sketch of this method is as follows

Step 0. Input: the matrix H,
the number first-order differential equations,
the time stepsize,
initial value,
start computation time;

end computation time;
the right function **f** and **F**.

Step 1. Calculation the exponential matrix **T** and the explicit and implicit coefficients $\mathbf{g}_j$, $\mathbf{g}_j^*$.
Step 2. Calculation the starting values $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4, \mathbf{F}_5, \mathbf{F}_6$, using second-order Runge–Kutta EF scheme.
Step 3. Calculation the approximation $\mathbf{y}_{k+1}$ using the seventh-step explicit EF scheme, i.e., Eq. (12) as predictor, then calculation $\mathbf{F}_{k+1}$ based on Eq. (4).
Step 4. Improvement this approximation using the seventh-step implicit EF scheme, i.e. Eq. (19), then upload $\mathbf{F}_{k+1}$ based on Eq. (4).

In Appendix A, we give the outline of C++ code which shows this computation.

3. Examples and numerical results

Example 1. As an illustrative example, we consider the Kepler problem. The Kepler problem describes the motion of two bodies which attract each other. For computing the motion of two bodies, we choose one of the bodies as the center of our coordinate system; the motion will then stay in a plane and we can use two-dimensional coordinates $q = (q_1, q_2)$ for the position of the second body. Newton's laws, with a suitable normalization, then yield the following differential equations

$$\ddot{q} = -\frac{q_1}{(q_1^2 + q_2^2)^{3/2}}, \quad \ddot{q} = -\frac{q_2}{(q_1^2 + q_2^2)^{3/2}}. \tag{29}$$

This is equivalent to a Hamiltonian system with the Hamiltonian

$$H(p_1, p_2, q_1, q_2) = \frac{1}{2}(p_1^2 + p_2^2) - \frac{1}{\sqrt{q_1^2 + q_2^2}}, \quad p_i = \dot{q}_i. \tag{30}$$

Every solution of (29) satisfies the total energy conservation and the angular momentum conservation

$$\frac{1}{2}(\dot{q}_1^2 + \dot{q}_2^2) - \frac{1}{\sqrt{q_1^2 + q_2^2}} = H_0, \quad q_1\dot{q}_2 - q_2\dot{q}_1 = L_0. \tag{31}$$

For the comparison, we take the same initial condition and the parameter values as [16]. The numerical schemes including the explicit Euler method, symplectic Euler method and implicit midpoint were applied to (29) (see [16]). They evaluated their schemes by comparing exact solutions with numerical solutions (see Fig. 1) and computing energy conservation. Here we use PCEF7 with a larger stepsize $h = 0.05$ to (29). The simulation is run for a long time of $t = 1200$. Eq. (29) is transformed equally to the following matrix form

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -p_1 - \frac{q_1}{(q_1^2+q_2^2)^{3/2}} \\ -p_2 - \frac{q_2}{(q_1^2+q_2^2)^{3/2}} \end{bmatrix}. \tag{32}$$

Fig. 2 shows our numerical solutions. Figs. 3 and 4 show the values $(H(p_n, q_n) - H(p_0, q_0))$ as a function of time ($H(p_0, q_0)$ is initial energy) by Ref. [16] and our method, respectively. Figs. 1 and 3 are copied from [16]. From above figures, it is seen the PCEF7 shows considerably greater accuracy than the numerical schemes Euler method, symplectic Euler method and implicit midpoint used in [16].

Example 2. We consider the interaction of seven argon atoms in a plane, where six of them are arranged symmetrically around a center atom (frozen argon crystal) (see Fig. 5). The total energy of the system is given by

$$H(p, q) = \frac{1}{2}\sum_{i=1}^{7} \frac{1}{m_i} p_i^{\mathrm{T}} p_i + \sum_{i=2}^{7} \sum_{j=1}^{i-1} V_{ij}(\|q_i - q_j\|), \tag{33}$$

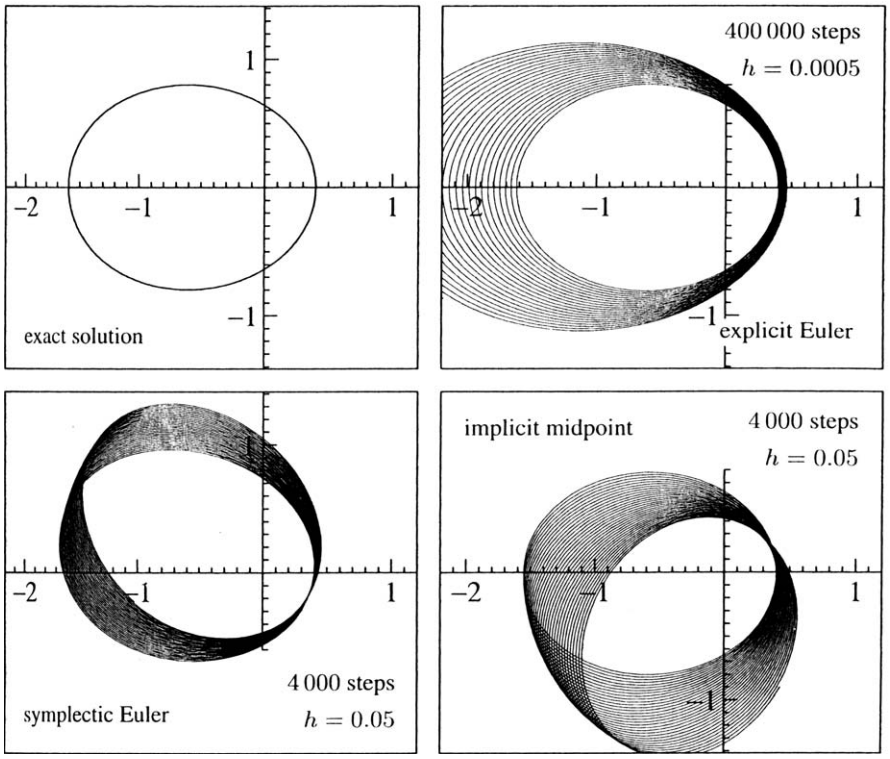

Fig. 1. Exact and numerical solutions of the Kepler problem.

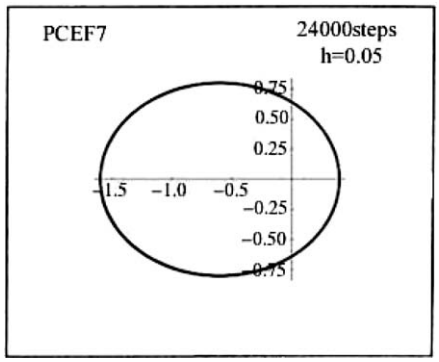

Fig. 2. Numerical solutions obtained by PCEF7.

where $V_{ij}(r)$ are given potential functions, here we choose the Lennard–Jones potential

$$V_{ij}(r) = 4\varepsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right),$$

where ε_{ij} and σ_{ij} are suitable constants depending on the atoms.

We take the same units as [16] for our calculations, masses in [kg], distances in nanometers, and times in nanoseconds. The initial conditions, the parameter values are given in Table 1, where $k_B = 1.380658 \times 10^{-23}$ [J/K] is Boltzmann's constant. The energy at the initial position is $H_0 = H(p_0, q_0) \approx -1260.2k_B$ [J].

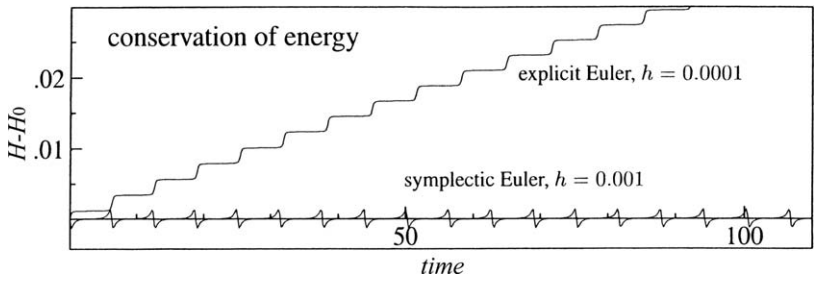

Fig. 3. Energy conservation.

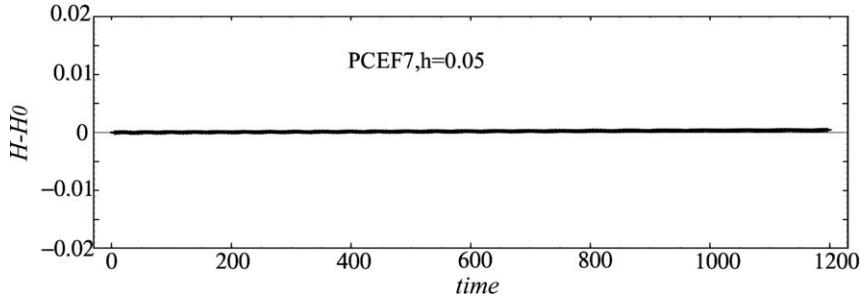

Fig. 4. Energy conservation obtained by PCEF7.

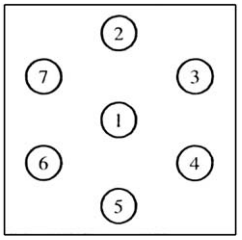

Fig. 5. Frozen argon crystal.

Table 1
Parameters and initial conditions for Example 2

Particle masses	$m_i = m = 66.34 \times 10^{-27}$ [kg]
Initial positions (nm)	$\mathbf{r}_1 = (0.0, 0.0)$, $\mathbf{r}_2 = (0.02, 0.39)$, $\mathbf{r}_3 = (0.34, 0.17)$, $\mathbf{r}_4 = (0.36, -0.21)$, $\mathbf{r}_5 = (-0.02, -0.40)$, $\mathbf{r}_6 = (-0.35, -0.16)$, $\mathbf{r}_7 = (-0.31, 0.21)$
Initial velocities (nm/nsec)	$\mathbf{v}_1 = (-30, -20)$, $\mathbf{v}_2 = (50, -90)$, $\mathbf{v}_3 = (-70, -60)$, $\mathbf{v}_4 = (90, 40)$, $\mathbf{v}_5 = (80, 90)$, $\mathbf{v}_6 = (-40, 100)$, $\mathbf{v}_7 = (-80, -60)$
ε_{ij}	$\varepsilon_{ij} = \varepsilon = 119.8 k_B$ [J]
σ_{ij}	$\sigma_{ij} = \sigma = 0.341$ [nm]

The system of differential equations associated with Eq. (33) is expressed as

$$\begin{cases} \dot{q}_i = \frac{p_i}{m_i} \\ \dot{p}_i = -\frac{\partial V_{ij}}{\partial q_i} \end{cases} \quad i = 1, 2, \ldots 7. \tag{34}$$

The numerical schemes including the explicit Euler method, symplectic Euler method and the Verlet method (see [16]) were applied to (34). The integrations are done over an interval of length 0.2 [nsec]. They evaluated their schemes by computing the error in the total energy (see Fig. 6), and they also calculated the temperature with the formula (35).

$$T = \frac{1}{2Nk_B}\sum_{i=1}^{7} m_i\|\dot{q}_i\|^2. \tag{35}$$

Here the non-matrix form PCEF7 is also applied to (34). Figs. 6 and 7 show the values $(H(p_n,q_n) - H(p_0,q_0))/k_B$ as a function of $t_n = nh$ given by Ref. [16] and our method, respectively. For the exact solution, this value is precisely zero for all times. Figs. 8 and 9 show the numerical values of the temperature difference $T - T_0$ with T given by $T_0 \approx 22.72$ [K] (initial temperature). In this problem, the temperature is not an exact invariant, and fluctuates around a constant value. Figs. 6 and 8 are copied from [16].

Here the non-matrix form PCEF7 is also applied to (34). Figs. 7 and 9 show energy conservation and temperature obtained by PCEF7, respectively. Both PCEF7 and the Verlet method need only one force evaluation per integration step, so the computational effort for the verlet and the PCEF7 algorithm is same. It is seen that PCEF7 shows the desired behavior.

Example 3. The third example is a one-dimensional simulation of three particles bound together by two springs of different stiffnesses (see Fig. 10).

The potential energy of the system is

$$V(x_1,x_2,x_3) = \frac{1}{2}k_1(|x_2 - x_1| - l_1)^2 + \frac{1}{2}k_2(|x_3 - x_2| - l_2)^2. \tag{36}$$

The parameters and initial conditions for this experiment are given in Table 2.

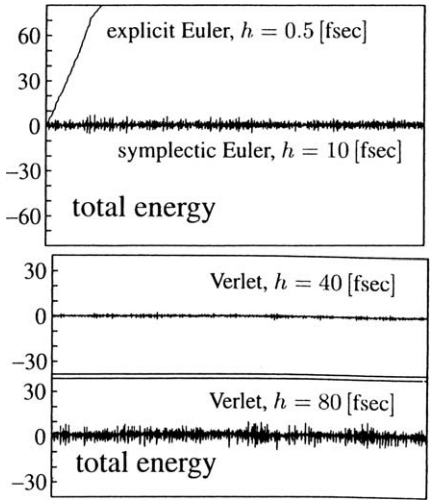

Fig. 6. The error in the total energy.

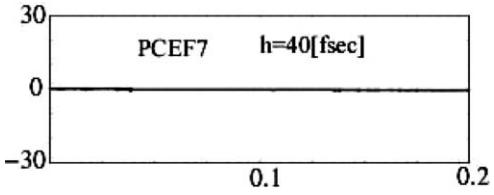

Fig. 7. The error in the total energy obtained by PCEF7.

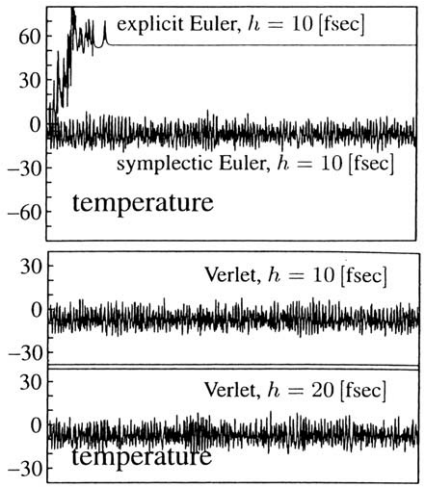

Fig. 8. Computed the temperature.

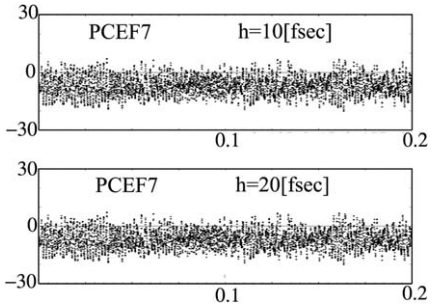

Fig. 9. Computed the temperature obtained by PCEF7.

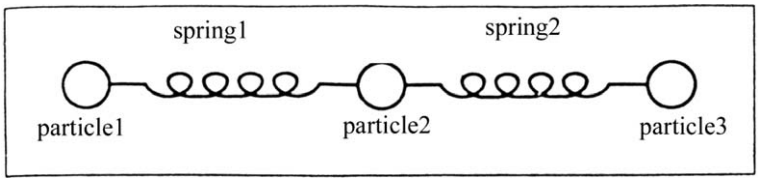

Fig. 10. Spring experiment, system of particles.

Table 2
Parameters and initial conditions for experiment

Particle masses	$m_1 = m_2 = m_3 = 1$
Initial positions	$x_1 = -7$, $x_2 = 0$, $x_3 = 7$
Initial velocities	All particles at rest
Spring stiffnesses	$k_1 = 16$, $k_2 = 1$
Equilibrium spring lengths	$l_1 = 6$, $l_2 = 6$

The initial total energy of the system is $H_0 = 8.5$. The numerical schemes including the Verlet-I, Verlet-II and Verlet-X defined in [17] were applied to this problem. The integrations were done over a long interval of length 2000. They evaluated their schemes by computing the total energy (see Figs. 11 and 12), and quantified the efficiency of their schemes by computing the number of force evaluations (see Fig. 14). Fig. 11 shows total energy for the three methods with a fixed stepsize $h = 0.01$, and Fig. 12 shows total energy for the Verlet-I method with a doubled stepsize $h = 0.02$. For Fig. 14, the average relative error in total energy (ξ) is computed with the formula

$$\xi = \frac{1}{s}\sum_{i=1}^{s}\frac{|H_i - H_0|}{|H_0|},$$

where s is again the number of samples taken and H_i is the total energy at the time of sample i. Figs. 11, 12 and 14 are copied from [17]. Here, we choose the matrix form PCEF7 to this problem, also integrate to 2000 and compare with [17]. Nine values of h are tested, geometrically increasing from $h = 0.005$ to 0.05. Fig. 13 shows total energy for the PCEF7 with a fixed stepsize $h = 0.05$. Fig. 15 shows the average relative error in total energy (ξ) versus number of force evaluations.

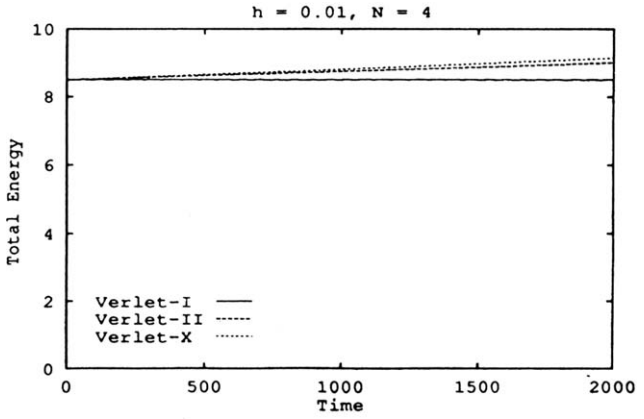

Fig. 11. Total energy for the three methods with a stepsize $h = 0.01$.

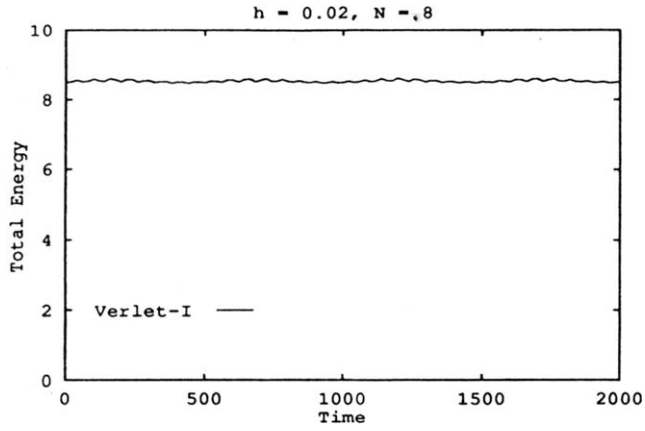

Fig. 12. Total energy for the Verlet-I method with a stepsize $h = 0.02$.

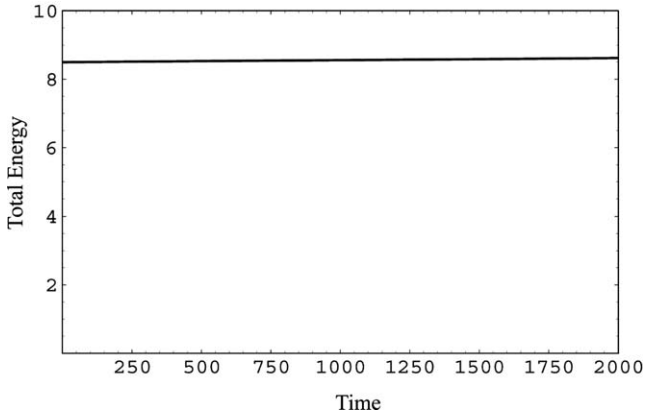

Fig. 13. Total energy for the PCEF7 with a stepsize $h = 0.05$.

The nonsymplectic methods Verlet-II and Verlet-X produce substantial energy drift after long-time integration, even when stepsize $h = 0.01$. The Verlet-II does not experience this problem, when stepsize $h = 0.02$. Our PCEF7 method conserves energy very well even when large steps are used ($h = 0.05$). From the

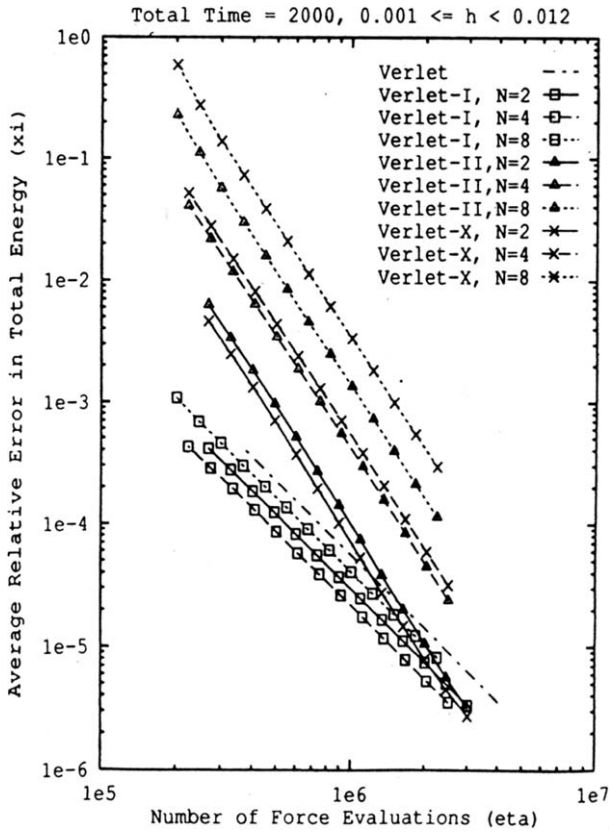

Fig. 14. The average relative error versus number of force evaluations [16].

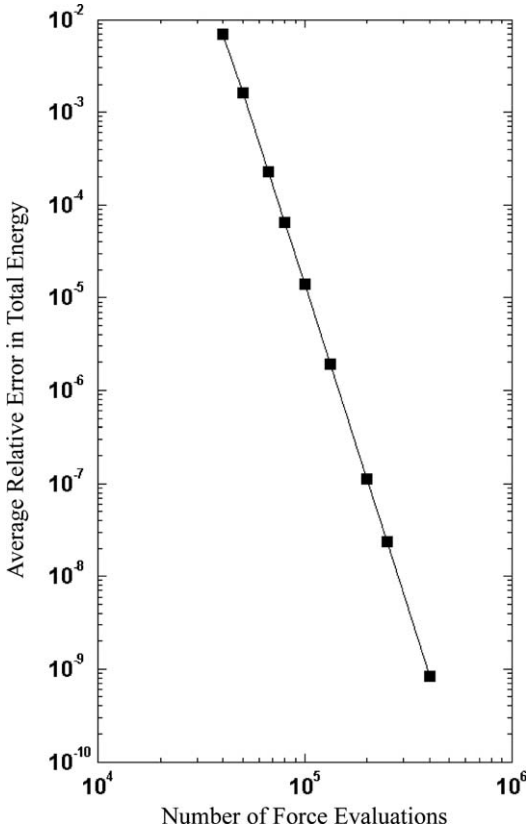

Fig. 15. The average relative error versus number of force evaluations obtained by PCEF7.

Figs. 11–15, it is seen the PCEF7 method shows considerably greater accuracy and efficiency than the numerical schemes the Verlet-I, Verlet-II and Verlet-X in [17].

4. Conclusion

We have presented the seventh-step explicit EF scheme and the seventh-step implicit EF scheme. The seventh-step explicit EF scheme and the seventh-step implicit EF scheme need only one force evaluation per integration step. We have applied the seventh-step predictor–corrector EF method to Kepler problem, the interaction of seven argon atoms in a plane and three particles bound together by two springs of different stiffnesses. The three problems have various potential functions. In all cases, the seventh-step predictor–corrector EF method gives desired results. The numerical results show that the scheme is very accurate and computationally efficient.

Acknowledgements

This work was supported by Liu Hui Applied Mathematics Center of Nankai-Tianjin University and the state key laboratory of scientific and engineering computing of the Chinese. In addition, the author express his gratitude to the anonymous reviewers whose comments and suggestions were most constructive, informative and useful during the revision of this manuscript. The author also thanks Prof. Jialin Hong in the state key laboratory of scientific and engineering computing of the Chinese for his kind help.

Appendix A

The following is the outline of C++ code which shows how to implement PCEF7.

```
void inputinit(      int*      n,    // the number first-order differential equations
                     int*      in,   // the step number of the scheme, here in=6
                     double*   t0,   // start computation time
                     double*   tn,   // end computation time
                     double*   h,    // the time stepsize
                     double**  H,    // the matrix H
                     double**  y0    // initial value
                ) //Input function

  void calF(
          double    t0,
          double*   y0,
          int       n,
          double*   F,   // the right function F
          ) // calculation the right function F

  void calf (
          double    t0,
          double*   y0,
          int       n,
          double*   f,   //   the right function f
          ) // calculation the right function f
```

There are functions for matrix operation, for example,

```
      void matadd ( ) // matrix add operation,
          ⋮
          ⋮
      void matinverse ( ) // calculation H^{-1},

  void calT ( ) //calculation T,

  void calgj ( ) // calculation explicit coefficients g_j,

  void calgj2 ( ) // calculation implicit coefficients g_j^*,

  void main( )
  {   while((tk < tn) && (k <= in)) // tk is present time
      { Calculation the starting values F_k using second-order Runge–Kutta EF
        scheme };
      while(tk < tn)
      { Calculation the approximation y_{k+1} using the seventh-step explicit EF scheme
        then calculation F_{k+1};
        Improvement this approximation using the seventh-step implicit EF scheme, then
        upload F_{k+1};
        output to file;
        tk= tk+h;}
  }
```

Note: Here we only provide main functions.

References

[1] R.B. Mann, D. Robbins, T. Ohta, Exact relativistic two-body motion in lineal gravity, Phys. Rev. Lett. 82 (1999) 3738.
[2] M. Quito, C. Monterola Jr., C. Saloma, Solving *N*-body problems with neural networks, Phys. Rev. Lett. 86 (2001) 4741–4744.
[3] J. Waltz, G.L. Page, S.D. Milder, J. Wallin, A. Antunes, A performance comparison of tree data structures for *N*-body simulation, Journal of Computational Physics 178 (2002) 1–14.

[4] R. Holland, Finite-difference time-domain (FDTD) analysis of magnetic diffusion, IEEE Trans. Electromagn. Compat. 36 (1994) 32–39.
[5] G. Beylkin, J.M. Keiser, L. Vozovoiy, A new class of time discretization chemes for the solution of nonlinear PDEs, J. Comput. Phys. 147 (1998) 362–387.
[6] B.J. McCartin, R.J. Mitchell, A numerical model of chaotic respiration, in: Proceedings of the Fourth Conference on Mathematical Modeling in the Undergraduate Curriculum, University of Wisconsin, La Crosse, 2000.
[7] B.J. McCartin, Exponential fitting of the delayed recruitment/renewal equation, J. Comput. Appl. Math. 136 (2001) 343.
[8] S.M. Cox, P.C. Matthews, Exponential time differencing for stiff systems, J. Comput. Phys. 176 (2002) 430.
[9] G.V. Berghe, L.G. Ixaru, H.D. Meyer, Frequency determination and step-length control for exponentially-fitted Runge–Kutta methods, J. Comput. Appl. Math. 132 (2001) 95.
[10] T.E. Simos, A fourth algebraic order exponentially-fitted Runge–Kutta method for the numerical solution of the SchrÖdinger equation, IMA J. Numer. Anal. 21 (2001) 919.
[11] G. Vanden Berghe, L.Gr. Ixarul, M. Van Daele, Optimal implicit exponentially-fitted Runge–Kutta methods, Comput. Phys. Commun. 140 (2001) 346.
[12] G. Beylkin, J.M. Keiser, L. Vozovoiy, A new class of time discretization schemes for the solution of nonlinear PDEs, J. Comput. Phys. 147 (1998) 362.
[13] L.Gr. Ixaru, G. Vanden Berghe, H. De Meyer, Exponentially fitted variable two-step BDF algorithm for first order ODEs, Comput. Phys. Commun. 150 (2003) 116.
[14] C. Tang, H. Yan, H. Zhang, W. Li, The various order explicit multistep exponential fitting for systems of ordinary differential equations, J. Comput. Appl. Math. 169 (2004) 171.
[15] C. Tang, H. Yan, H. Zhang, Z. Chen, M. Liu, G. Zhang, The arbitrary order implicit multistep schemes of exponential fitting and their application, J. Comput. Appl. Math. 173 (2005) 155.
[16] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration, Springer-Verlag, Berlin, Heidelberg, 2002.
[17] J.J. Biesiadecki, R.D. Skeel, Dangers of multiple time step methods, J. Comput. Phys. 109 (1993) 318.