High-order schemes for conservative or dissipative systems

Takayasu Matsuo

Department of Computational Science and Engineering, Graduate School of Engineering, Nagoya University, Nagoya, 464-8603, Japan

Received 30 November 2001; received in revised form 30 May 2002

Abstract

We propose a new method for designing high-order finite difference schemes that inherit conservation or dissipation properties from conservative or dissipative systems such as Hamiltonian systems with/without damping terms. The proposed method has a feature that the computational costs of the resulting schemes do not increase in practice, even when the order of accuracy is increased.

Keywords: Finite difference method; Energy conservation; Energy dissipation; Hamiltonian systems; Backward difference formula

1. Introduction

In the present paper, we consider a numerical method for the system

$$\frac{d}{dt} z(t) = A \nabla H(z), \quad t > 0,$$

$$z(0) = z_0,$$

where $z : \mathbb{R} \to \mathbb{R}^N$, $A$ is an $N \times N$ real matrix, $H : \mathbb{R}^N \to \mathbb{R}$, $\nabla H(z)$ is the gradient of $H$ with respect to $z$, and $z_0 \in \mathbb{R}^N$ is a given initial value. When $A$ is skew-symmetric, $H$ is preserved along the solution:

$$\frac{d}{dt} H(z(t)) = (\nabla H)^T \dot{z} = (\nabla H)^T A \nabla H = 0,$$

$\star$ Supported in part by a Grant-in-Aid for Scientific Research (B) of the Ministry of Education, Culture, Sports, Science and Technology of Japan, and by “Research for the Future Program” of the Japan Society for the Promotion of Science.

E-mail address: matsuo@na.cse.nagoya-u.ac.jp (T. Matsuo).
where $\dot{z} = (d/dt)z(t)$ and $\cdot^T$ denotes transpose of the matrix. Therefore, in this case, system (1) is called conservative. Hamiltonian systems, in which $N = 2M$ ($M \in \mathbb{N}$) and

$$A = \begin{pmatrix} 0 & -I_M \\ I_M & 0 \end{pmatrix}, \quad I_M: \text{identity matrix of order } M$$

is an example of the conservative system. When $A$ is negative definite, $H$ decreases along the solution:

$$\frac{d}{dt} H(z(t)) = (\nabla H)^T A \nabla H \leq 0$$

and hence system (1) is called dissipative. Hamiltonian systems having a damping term, in which

$$A = \begin{pmatrix} -\alpha I_M & -I_M \\ I_M & 0 \end{pmatrix}, \quad \mathbb{R} \ni \alpha > 0,$$

is an example of the dissipative system. We also encounter these kinds of equations as a result of spatial discretization of certain conservative or dissipative partial differential equations, such as the KdV equation, the nonlinear Schrödinger equation, or the Cahn–Hilliard equation.

For the conservative or dissipative systems, numerical schemes that inherit the conservation or dissipation property, which schemes are referred to as conservative or dissipative schemes in the present paper, are desirable. In the 1970s and 1980s, associated with specific problems, several conservative or dissipative schemes of first- or second-order accuracy were presented (e.g. [1,6]). Then, in the 1990s, general methods for designing conservative or dissipative schemes of first- or second-order accuracy have been independently proposed by McLachlan et al. [11] (see also Gonzalez [5]), and Furihata and Matsuo [2–4,9,10]. These two methods have one key concept in common: the “discrete derivative”, or “discrete gradient”, by which we define a numerical scheme analogously to the original system (1) so that the conservation or dissipation property results.

Of higher order schemes, however, our knowledge has been limited. In the conservative case, we can construct higher order conservative schemes by the so-called composition technique [12], based on the second-order conservative scheme derived by the general methods mentioned above. However, this technique has a drawback that the computational cost of the resulting schemes increases heavily as the order of accuracy is increased. In the dissipative case, we have not had a general method for constructing high-order dissipative schemes at all. In the present paper, we here propose a new general method, for both conservative and dissipative cases, by which we can construct high-order conservative or dissipative schemes. The method has an advantage that the computational costs of the resulting schemes do not increase in practice, even when the order of accuracy is increased. The present paper is organized as follows. In Section 2 we briefly review the general methods for constructing second-order conservative or dissipative schemes and the composition technique. In Section 3 we propose a new method for constructing high-order schemes and present several numerical examples. Finally, Section 4 presents concluding remarks.

2. Second-order conservative or dissipative schemes and the composition method

In this section, we review the general methods for designing second-order conservative or dissipative schemes [2,5,9,11], together with the “composition” method for obtaining higher-order schemes [12].
2.1. Second-order conservative or dissipative schemes

The general methods for designing second-order conservative or dissipative schemes are based on the concept of the “discrete derivative”, or the “discrete gradient”. Using this concept, we define schemes analogously to the original system (1), so that the conservation or dissipation property results.

Let $\Delta t$ be a time-mesh size. We denote numerical solutions by $z^{(m)} \approx z(m\Delta t)$. In the present paper, $H(z^{(m)})$ is often abbreviated as $H^{(m)}$. Then, the discrete derivative is defined as follows.

**Definition 2.1** (First- or second-order discrete derivative). 1 For a sufficiently smooth function $f: \mathbb{R}^N \rightarrow \mathbb{R}$, we call $\nabla_d f: \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ “a discrete derivative” if it satisfies the following two conditions.

1. $f(y_1) - f(y_2) = (\nabla_d f(y_1, y_2))^T (y_1 - y_2)$ for all $y_1, y_2 \in \mathbb{R}^N$.
2. For any sufficiently smooth function $x(t): \mathbb{R} \rightarrow \mathbb{R}^N$, any $t_1, t_2 \in \mathbb{R}$, and $p = 1$ or 2, there exist $\tilde{t}(t_1, t_2)$ such that $\nabla_d f(x(t_1), x(t_2)) = \nabla f(x(\tilde{t})) + O(|t_1 - t_2|^p)$ holds.

Furthermore, we call $\nabla_d f$ “a first-order discrete derivative at $\tilde{t}$” when $p = 1$, and a “second-order discrete derivative at $\tilde{t}$” when $p = 2$.

The first condition in Definition 2.1 implies that the discrete derivative should satisfy a discrete version of the chain rule $df = (\nabla f)^T dy$. The second condition defines the order of the discrete derivative. Here, in order to give a flavor of the definition, we show an simple example. When $N=1$, $\nabla_d f(y_1, y_2) = (f(y_1) - f(y_2))/(y_1 - y_2)$ is a second-order discrete derivative at $\tilde{t} = (t_1 + t_2)/2$. In fact, this $\nabla_d f$ obviously satisfies condition (1), and since

$$\nabla_d f(x(t_1), x(t_2)) = \frac{f(x(t_1)) - f(x(t_2))}{x(t_1) - x(t_2)} = \nabla f \left( x \left( \frac{t_1 + t_2}{2} \right) \right) + O((t_1 - t_2)^2),$$

it also satisfies condition (2) for $\tilde{t} = (t_1 + t_2)/2$ and $p = 2$.

Using the above discrete derivative, we can design conservative or dissipative schemes as follows.

**Scheme 1** (First- or second-order scheme). A scheme

$$\frac{z^{(m+1)} - z^{(m)}}{\Delta t} = A \nabla_d H(z^{(m+1)}, z^{(m)}) \quad (m = 1, 2, \ldots),$$

$$z^{(0)} = z_0 \quad (an \ initial \ value)$$

is conservative:

$$H^{(m)} = H^{(0)} \quad (m = 1, 2, \ldots)$$

1 The second-order discrete derivatives and schemes are of primary interest here. However, descriptions for first-order discrete derivatives and schemes are also provided for the sake of completeness.
if \( A \) is skew-symmetric, or is dissipative:
\[
H^{(m+1)} \leq H^{(m)} \quad (m = 1, 2, \ldots)
\]
(6)
if \( A \) is negative definite. The scheme is second-order, if the discrete derivative \( \nabla_dH \) is second-order at \( \tilde{t} = ((m + 1)\Delta t + m\Delta t)/2 \). Otherwise, the scheme is first-order.

**Proof.** In light of the definition of \( A \), if the scheme is second-order for \( t \), then conservation (5) or dissipation (6) is clear. In addition, since the left-hand side of (4) is negative definite, the scheme is second-order. The claim on the order of the scheme is also clear.

Several kinds of concrete forms of discrete derivative are known, which are given in the following.

**Itoh and Abe’s discrete derivative:** Itoh and Abe proposed the following derivative [8].

\[
\nabla_d f(x, y) = f(y_1, \ldots, y_{i-1}, x_i, x_{i+1}, \ldots, y_N) - f(y_1, \ldots, y_{i-1}, y_i, x_{i+1}, \ldots, y_N) \frac{x_i - y_i}{x_i - y_i},
\]
(7)
where \( 1 \leq i \leq N \). This discrete derivative is first-order at any \( \tilde{t} \in [t_1, t_2] \).

**Gonzalez’s discrete derivative:** Gonzalez proposed the following derivative [5].

\[
\nabla_d f(x, y) = f'(z) + \frac{f(x) - f(y) - (f'(z))(x - y)}{\|x - y\|_2} (x - y),
\]
(8)
where \( \|\cdot\|_2 \) is the Euclidean norm. We can prove by simple calculation that this discrete derivative is of order two at \( \tilde{t} = (t_1 + t_2)/2 \) (see Definition 2.1). However, this discrete derivative has a disadvantage in that all of the variables couple each other due to the term \( \|x - y\|_2^2 \).

**Furihata and Matsuo’s discrete derivative:** Based on the trivial identity:
\[
f(x_1, x_2) - f(y_1, y_2) = \frac{f(x_1, x_2) - f(y_1, x_2) + f(x_1, y_2) - f(y_1, y_2)}{2}
+ \frac{f(x_1, x_2) - f(x_1, y_2) + f(y_1, x_2) - f(y_1, y_2)}{2}.
\]
(9)
Furihata and Matsuo proposed a method by which to construct a discrete derivative [2,9]. In the two-dimensional case, the discrete derivatives are given by
\[
\nabla_d f(x, y)_1 = \frac{f(x_1, x_2) - f(y_1, x_2) + f(x_1, y_2) - f(y_1, y_2)}{2(x_1 - y_1)},
\]
(10)
\[
\nabla_d f(x, y)_2 = \frac{f(x_1, x_2) - f(x_1, y_2) + f(y_1, x_2) - f(y_1, y_2)}{2(x_2 - y_2)}.
\]
(11)
In the $N \geq 3$-dimensional case, we can also construct a discrete derivative, by repeatedly applying the identity. The discrete derivative constructed in this way is second-order at $\tilde{t} = (t_1 + t_2)/2$ (see Definition 2.1). A drawback of these discrete derivatives is that the symmetry with respect to $x_1, \ldots, x_N$ (thus also to $y_1, \ldots, y_N$) is destroyed.

2.2. The composition method

We can easily construct higher-order schemes based on the above described schemes. Let us rewrite scheme (1) as $z^{(m+1)} = \phi(\Delta t)z^{(m)}$, where $\phi(\Delta t) : \mathbb{R}^N \to \mathbb{R}^N$ is the map that evolves $z^{(m)}$ into $z^{(m+1)}$, and assume that $\phi(\Delta t)$ is time-symmetric: $\phi(-\Delta t) = \phi^{-1}(\Delta t)$. This means that the discrete derivative should be time-symmetric, which is the case for the discrete derivatives of Gonzalez and Furihata–Matsuo. Note that the scheme is then automatically second-order. A fourth-order scheme can then be obtained by the composition:

$$
\phi_4(\Delta t) = \phi(c_1 \Delta t)\phi(c_2 \Delta t)\phi(c_1 \Delta t),
$$

$$
c_1 = 1/(2 - \sqrt{2}) \simeq 1.35, \quad c_2 = 1 - 2c_1 \simeq -1.70.
$$

That is, we simply apply the second-order scheme thrice in succession in one time step using different time mesh sizes. In addition, a sixth-order scheme can be obtained by $^2$

$$
\phi_6(\Delta t) = \phi(c_1 \Delta t)\phi(c_2 \Delta t)\phi(c_3 \Delta t)\phi(c_4 \Delta t)\phi(c_3 \Delta t)\phi(c_2 \Delta t)\phi(c_1 \Delta t),
$$

$$
c_1 = 0.784513610477560, \quad c_2 = 0.235573213359357, \quad c_3 = -0.117767998417887, \quad c_4 = 1 - 2(c_1 + c_2 + c_3).
$$

In a similar manner, we can construct any high-order scheme by appropriately increasing the number of compositions in one time step. This technique, called the “composition technique” [12], is very convenient and often quite practical. However, this method has two drawbacks. First, the computational cost becomes approximately three times (seven times) larger for the fourth-order (sixth-order) scheme. Secondly, when applied to dissipative systems, the dissipation property might be lost due to the negative time steps (e.g., $-1.7\Delta t$ in (12)). Thus, in the present paper we examine another possible method for constructing high-order schemes.

3. New high-order conservative or dissipative schemes

In this section, we present a new method for designing high-order conservative or dissipative schemes. In addition, we present several numerical examples.

3.1. High-order conservative or dissipative schemes

The newly proposed method is based on a simple idea. Scheme 1 was first- or second-order because both sides of the scheme were first- or second-order and no special care was taken to

---

$^2$ The coefficients $c_i$ are determined by solving a certain system of nonlinear equations [12]. Here, we list only the approximate values.
improve the accuracy of this scheme. Therefore, let us consider making both sides of Scheme 1 high-order, while, at the same time, maintaining the conservation or dissipation property. This idea can be realized in fact. We first introduce \( p \)-th order difference operators as follows.

**Definition 3.1** (\( p \)-th order difference operator). Let \( \delta^{(1)}_{m,c} \) be a difference operator defined as

\[
\delta^{(1)}_{m,c} z^{(m)} \equiv \sum_{i=-l_1}^{l_2} c_i z^{(m+i)} / \Delta t,
\]

(14)

where \( l_1, l_2 \in \mathbb{N}, c_i \in \mathbb{R}, \) and \( c = (c_{-l_1}, \ldots, c_{l_2}). \) If for any sufficiently smooth function \( x(t) : \mathbb{R} \to \mathbb{R}^N \) and any \( t \in \mathbb{R}, \) there exists \( \tilde{t}(t, \Delta t, l_1, l_2) \) such that

\[
\sum_{i=-l_1}^{l_2} c_i x(t + i\Delta t) / \Delta t = \dot{x} (\tilde{t}) + O(\Delta t^p)
\]

(15)

holds, we call \( \delta^{(1)}_{m,c} \) “a \( p \)-th order difference operator at \( \tilde{t} \),” and denote it by \( \delta^{(1),p}_{m,c}. \)

The symbol \( \delta^{(1)}_{m,c} \) is too complicated. Hence the subscript \( c \) is often omitted where no confusion occurs. An example of \( p \)-th order difference operator is \( \delta^{(1)}_m z^{(m)} = (-\frac{1}{24} z^{(m+2)} + \frac{9}{8} z^{(m+1)} - \frac{9}{8} z^{(m)} + \frac{1}{24} z^{(m-1)}) / \Delta t, \) which is, in fact, a fourth-order difference operator at \( \tilde{t} = (m + 1/2) \Delta t. \)

Next, we define a high-order discrete derivative as follows.

**Definition 3.2** (\( p \)-th order discrete derivative). Let \( \delta^{(1),p}_{m,c} \) a \( p \)-th order difference operator at \( \tilde{t}. \) For a sufficiently smooth function \( f : \mathbb{R}^N \to \mathbb{R}, \) we call

\[
\nabla^P f : \mathbb{R}^N \times \cdots \times \mathbb{R}^N \to \mathbb{R}^N
\]

“a \( p \)-th order discrete derivative (corresponding to \( \delta^{(1),p}_{m,c} \))”, if the function satisfies the following conditions.

(1) \( \delta^{(1),p}_{m,c} f (y^m) = (\nabla^P f (y_{m-l_1}, \ldots, y_{m+l_2}))^T \delta^{(1),p}_{m,c} y^m \) for any \( y_{m-l_1}, \ldots, y_{m+l_2} \in \mathbb{R}^N, \)

(2) For any sufficiently smooth function \( x(t) : \mathbb{R} \to \mathbb{R}^N \) and any \( t \in \mathbb{R}, \) \( \nabla^P f (x(t - l_1 \Delta t), \ldots, x(t + l_2 \Delta t)) = \nabla f (x(\tilde{t})) + O(\Delta t^p) \) holds.

The following theorem gives us a concrete form of the discrete derivative.

**Theorem 3.3.** Let \( \delta^{(1),p}_{m,c} \) be a \( p \)-th order difference operator at \( \tilde{t}. \) Then, the following is a \( p \)-th order discrete derivative.

\[
\nabla f (\tilde{y}) + \frac{\delta^{(1),p}_{m,c} f (y^m) - \nabla f (\tilde{y})^T \delta^{(1),p}_{m,c} y^m}{\| \delta^{(1),p}_{m,c} y^m \|^2_2} \delta^{(1),p}_{m,c} y^m,
\]

(16)

where \( \tilde{y} = \tilde{y}(y_{m-l_1}, \ldots, y_{m+l_2}) \) is a function of \( y_{m-l_1}, \ldots, y_{m+l_2} \in \mathbb{R}^N \) such that for any sufficiently smooth function \( x(t) : \mathbb{R} \to \mathbb{R}^N \) and any \( t \in \mathbb{R}, \) \( \tilde{y}(x(t - l_1 \Delta t), \ldots, x(t + l_2 \Delta t)) = x(\tilde{t}) + O(\Delta t^p) \) holds.
Proof. The first condition in Definition 3.2 is trivial from (16). In order to prove the second condition, we substitute $x(t-l_1\Delta t),\ldots,x(t+l_2\Delta t)$ into $y_{m-l_1},\ldots,y_{m+l_2}$ and expand each term in (16). From the definition of $\delta^{(1),p}_{m\ell} f(x(t+m\Delta t)) = f(x(\tilde{t})) + O(\Delta t^p)$ and $\delta^{(1),p}_{m\ell} x(t+m\Delta t) = \dot{x}(\tilde{t}) + O(\Delta t^p)$. In addition, from the definition of $\tilde{y}$, we have $\nabla f(\tilde{y}) = \nabla f(x(\tilde{t}))/SO_{\langle 1 \rangle; p m}$ and $\nabla x(\tilde{t})/SO_{\langle 1 \rangle; p m}$.

Thus, (16) is evaluated as

$$\nabla f(x(\tilde{t})) + O(\Delta t^p) + \frac{\dot{f}(x(\tilde{t})) + O(\Delta t^p)}{\|\dot{x}(\tilde{t}) + O(\Delta t^p)\|^2_2}(\dot{x}(\tilde{t}) + O(\Delta t^p)) = \nabla f(x(\tilde{t})) + O(\Delta t^p).$$

This implies the second condition of Definition 3.2. □

The discrete derivative (16) can be interpreted as follows. The first term is the “true” derivative at $y = \tilde{y}$, and the second term is the correction term of $O(\Delta t^p)$ that enforces the discrete chain rule (1) in Definition 3.2.

Using the discrete derivative, we can construct $p$th-order conservative or dissipative schemes for (1). The proof of the properties (the conservation or the dissipation, and the order) of the schemes is straightforward and hence omitted.

Scheme 2 ($p$th-order scheme). Let $\delta^{(1),p}_{m\ell}$ be a $p$th-order difference operator at $\tilde{t}$, and let $\nabla^p \partial H$ be the $p$th-order discrete derivative of $H$ at $\tilde{t}$. Then, a scheme

$$\delta^{(1),p}_{m\ell} z^{(m)} = A \nabla^p \partial H(z^{(m-l_1)},\ldots,z^{(m+l_2)}), \quad (m = l_1, l_1 + 1, \ldots),$$

$$z^{(0)} = z_0 \quad (an \ initial-value),$$

$$z^{(1)},\ldots,z^{(l-2)} \quad (starting-values)$$

is conservative:

$$\delta^{(1),p}_{m\ell} H^{(m)} = 0 \quad (m = l_1, l_1 + 1, \ldots), \quad \text{if } A \text{ is skew-symmetric}$$

or is dissipative:

$$\delta^{(1),p}_{m\ell} H^{(m)} \leq 0 \quad (m = l_1, l_1 + 1, \ldots), \quad \text{if } A \text{ is negative definite}.$$  

In addition, the scheme is $p$th-order.

This scheme is not self-starting. We have to provide the starting-values $z^{(1)},\ldots,z^{(l-2)}$ in addition to the initial value $z^{(0)} = z_0$. Usually, these are computed using another scheme with a sufficiently accurate, such as Scheme sch1 or the Runge–Kutta methods.

Since this scheme is a system of nonlinear equations (17), we must solve it at each time step. Note that the only unknown variable is $z^{(m+l_2)}$, and the rest of variables ($z^{(m-l_1)},\ldots,z^{(m+l_2-1)}$) are all known “previous” values. Commonly, Newton’s method is used to obtain $z^{(m+l_2)}$. However, in
practice, use of the following simple iteration, which is derived from (17), is sufficient:

\[ \mathbf{z}^{(m+l_2,j)} = \frac{1}{d_{l_2}} \left( -\sum_{i=-l_1}^{l_2-1} d_i \mathbf{z}^{(m+i)} + \Delta t A \nabla_d^p H(\mathbf{z}^{(m-l_1)}, \ldots, \mathbf{z}^{(m+l_2-1)}, \mathbf{z}^{(m+l_2,j-1)}) \right), \]  

(20)

where \( j=1,2,\ldots \) is the index of iteration. In the numerical examples of the next section, we employed this simple iteration.

We should like to note that we can expect that the computational cost of the scheme based on iteration (20) will not increase in practice, even when the order of accuracy \( p \) is increased. This can be understood in the following way. The primary part of the computational cost is the cost of solving the system of nonlinear equations by iteration (20). This cost depends on (i) the number of the unknowns, (ii) the number of iterations required in (20), and (iii) the cost required for each iteration. We examine (i)–(iii) in turn. (i) As mentioned above, the number of unknowns remains constant at \( N \) regardless of the order of accuracy \( p \). (ii) Although the number of iterations is difficult to estimate theoretically, if we carefully choose \( \delta_m^{(1),p} \) and \( \Delta t \) so that the resulting scheme is sufficiently numerically stable, we can expect that the number of iterations remains to some level when \( p \) increases. The numerical results presented in the next section support this view (see Tables 1 and 2). (iii) The cost of each iteration increases slightly as \( p \) increases, because a few additional calculations are needed in the evaluation of the discrete derivative (16). However, this increase is rather moderate. By taking (i)–(iii) all into account, we thus know that even when the order of accuracy \( p \) is increased, the increase in the computational cost remains quite moderate, or even negligible, in a practical sense.
Remark 3.4. The statement in Scheme 2 holds for any $\delta_m^{(1),p}$. However, from a practical standpoint, $\delta_m^{(1),p}$ should be chosen so that the resulting schemes become numerically stable, which indicates $\delta_m^{(1),p}$ to be chosen as, for example, the stiffly stable “backward-difference operators” [7].

3.2. Numerical examples: (damped) Kepler problem

We now present a few numerical examples. Here, we examine the Kepler problem with/without a damping term.

**Problem 3.5 ((Damped) Kepler problem).** Let $N = 4$ and $H(z) = (z_1^2 + z_2^2)/2 - 1/\sqrt{z_3^2 + z_4^2}$. Let also $\alpha \in \mathbb{R}$ (damping factor) be a positive constant, and let $\varepsilon = 0.8$ (eccentricity). Then, find the solution to the initial-value problem:

$$\frac{dz}{dt} = \left( \begin{array}{cc} -\alpha I_2 & -I_2 \\ I_2 & 0 \end{array} \right) H'(z), \quad t > 0,$$

$$z(0) = (0, \sqrt{(1 + \varepsilon)(1 - \varepsilon)}, 1 - \varepsilon, 0)^T$$

for either the conservative case (C): $\alpha = 0$, or the dissipative case (D): $\alpha > 0$.

We compare the following numerical schemes:

- **PROPp**: The proposed Scheme 2, where $\delta_m^{(1),p}$ are the backward difference operators ($p = 2, 3, 4, 5, 6$);
- **COMPp**: The conventional second-order Scheme 1 with the discrete derivative of (8), along with its fourth- and sixth-order compositions ($p = 2, 4, 6$);
- **RK4**: The classical fourth-order Runge–Kutta scheme.

In the computations of (PROPp) and (COMPp), we used the simple iteration of (20). For (COMPp), this simplifies to

$$z^{(m+1),j} = z^{(m),j} + \Delta t A \nabla_d H(z^{(m),j}, z^{(m+1),j}), \quad j = 0, 1, 2, \ldots .$$

**Conservative Case (C):** Fig. 1 shows the accuracy of the numerical solutions at $t = T = 10$. We denote the number of mesh points by $N_t$ (thus $\Delta t = T/N_t$). The vertical axis is the absolute error defined by $\|z^{(N)} - z(T)\|_{\infty}$. The numerical results are plotted using points, and the lines are guides showing the $p$th-order lines, which are for the convenience of the reader. The figure shows that the proposed schemes (PROPp) are in fact of order $p$.

Fig. 2 shows the evolution of the energy $H$, which in the conservative case should be strictly preserved. Here, we compare the three fourth-order schemes: (RK4) and (PROP4) with $N_t = 150$, and (COMP4) with $N_t = 300$ (for (COMP4), $N_t = 150$ was too small for iteration (20) to work). This reveals another disadvantage of the fourth-order composition (12), namely that the fourth-order composition involves a larger time-mesh size, $-1.7\Delta t$, than originally required $\Delta t$. Note that the conservation property (18) for the proposed schemes (PROPp) involves $l$ steps, and hence does not immediately result in a local conservation property such as (5) for the scheme (COMP2) (and thus for (COMP4) and (COMP6)). However, if the initial- and starting-values of (PROPp) satisfy the condition: $H^{(0)} = \cdots = H^{(l-2)}$, then (18) implies the local conservation property $H^{(m)} = H^{(0)}$.
Based on Fig. 2, we observe that the solutions obtained by (PROP4) and (COMP4) correctly preserve the energy. In contrast, the solution obtained by (RK4) monotonically dissipates the energy and thus becomes less reliable over time.

Table 1 shows the total costs of the conservative schemes (PROPp) and (COMPp) for several $N_t$. Numbers in the table represent the total number of iterations of (20) from beginning to end, which can be used as a measure of computational cost. As expected, (COMP4) requires a cost which is approximately three times larger than that required for (COMP2), and (COMP6) requires a cost
which is approximately seven times larger than that required for (COMP2). On the other hand, the costs for (PROP2),(PROP4) and (PROP6) are approximately the same, and thus these schemes are far faster than the composed schemes. In other words, (PROP6) is the cheapest conservative scheme for the problem.

Thus, in the conservative case, we conclude that the proposed conservative Scheme 2 is superior to the conventional schemes with respect to reliability and computational cost.

**Dissipative Case** (D): Here, we take $\varepsilon = 0.01$. Fig. 3 shows the accuracy of the numerical solutions at $t = T = 10$. As before, the proposed scheme (PROPp) is correctly of order $p$.

Fig. 4 shows the evolution of the energy $H$. In this case, the energy should decrease monotonically. We now compare the fourth-order schemes (RK4) and (PROP4) with $N_t = 150$ and (COMP4) with $N_t = 300$. As in the conservative case, the dissipation property (19) for (PROPp) does not immediately result in a local dissipation property such as (6) for (COMP2). Rather, for the fourth-order backward-difference operator, for example:

$$
\delta_m^{(1),4} z^{(m)} = \frac{1}{\Delta t} \left( \frac{25}{12} z^{(m)} - 4z^{(m-1)} + 3z^{(m-2)} - \frac{4}{3} z^{(m-3)} + \frac{1}{4} z^{(m-4)} \right),
$$

(22)

we have a more relaxed result:

$$
z_i^{(m)} \leq \frac{48}{25} z_i^{(m-1)} - \frac{36}{25} z_i^{(m-2)} + \frac{16}{25} z_i^{(m-3)} - \frac{3}{25} z_i^{(m-4)}
$$

**Fig. 3.** Accuracy of the dissipative schemes.
The coefficient vector in \( \{ \cdot \} \) is bounded and rapidly tends toward a constant vector \((25/12, -23/12, 13/12, -3/12)\). Thus, practically speaking, the dissipation property is maintained, and in particular, the numerical solution is bounded. In fact, based on Fig. 1, the proposed scheme (PROP4) correctly dissipates the energy. Furthermore, the result agrees quite well with the exact result. The scheme (RK4) also dissipates the energy, but the profile is quite different from the exact profile and so is not at all satisfactory. The scheme (COMP4) works quite well for the problem, and it gives a result as good as that of (PROP4). This is rather unexpected, because the dissipation property can be lost in the composition. Whether this is the case for other dissipative problems in not known.

Table 2 shows the total number of iterations for (PROPp) and (COMPP). As in the conservative case, the cost of the composed schemes increases remarkably as the order is increased, whereas the cost of the proposed schemes (PROPp) remains almost constant. Therefore, we conclude that (PROP6) is the cheapest dissipative scheme for the problem.

Thus, we reach the same conclusion as in the conservative case. The proposed scheme is more reliable and computationally far less expensive than existing methods.

4. Concluding remarks

In the present paper, we have proposed a new method for designing high-order finite difference schemes that inherit the conservation or dissipation property from conservative or dissipative systems (1). The proposed method has a feature that the practical computational costs of the resulting schemes do not increase even when the order of accuracy \( p \) is increased. The numerical examples for the Kepler problem with/without a damping term show that the newly proposed schemes are in fact highly accurate, and more reliable and far cheaper than conventional numerical methods, such as the Runge–Kutta method and high-order schemes obtained by composition.
We would like to comment further on the proposed method. First, the time-mesh size control technique can be used when we implement the schemes derived by the proposed method, without losing the properties of the schemes (i.e., the dissipation or conservation property and the order). Secondly, the concept proposed in the present paper is also useful for designing numerical schemes of conservative or dissipative partial differential equations. If the space variables can be appropriately discretized so that the resulting system of ordinary differential equations retains an energy conservation or dissipation property (this is possible in many cases (see [2,9,10])), simply applying the proposed method to the system will yield a fully discrete conservative or dissipative scheme that is high-order with respect to time.

Acknowledgements

The author would like to express his gratitude for the helpful discussions with Dr. D. Furihata. The author also would like to thank Prof. M. Sugihara and Prof. M. Mori for their kind advice and helpful suggestions.

References