MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Arbitrary High-order Conservative or Dissipative Method for Ordinary Differential Equations and Its Block Algorithm

Takayasu MATSUO

METR 2005–19 July 2005

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Arbitrary high-order conservative or dissipative method for ordinary differential equations and its block algorithm*

Takayasu Matsuo†

July 25, 2005

Abstract

A method for designing arbitrary high-order conservative/dissipative finite difference schemes for conservative/dissipative ordinary differential equations is proposed. The new method is given by extending the previously presented method for up to the sixth order schemes. The key in this extension is to introduce the idea of the generalized backward difference formula (GBDF), by which the seventh- or higher-order schemes are stabilized. A "block" algorithm to reduce the computational cost is also presented. Numerical examples are shown to show the effectiveness of the method.

1 Introduction

We consider numerical solution to the system:

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{z}(t) = A\nabla H(\boldsymbol{z}), \quad t > 0, \\ \boldsymbol{z}(0) = \boldsymbol{z}_0, \end{cases} \tag{1}$$

where $\boldsymbol{z} : \mathbf{R} \to \mathbf{R}^N$, A is an $N \times N$ real matrix, $H : \mathbf{R}^N \to \mathbf{R}$, $\nabla H(z)$ is the gradient of H with respect to z, and $\boldsymbol{z}_0 \in \mathbf{R}^N$ is a given initial value. When A is skew-symmetric, H is preserved along the solution:

$$\frac{\mathrm{d}}{\mathrm{d}t}H(\boldsymbol{z}(t)) = (\nabla H)^{\mathrm{T}}\,\dot{\boldsymbol{z}} = (\nabla H)^{\mathrm{T}}\,A\nabla H = 0, \tag{2}$$

*This work is partially supported by the Grant-in-Aid for Scientific Research (B) of the Ministry of Education, Science, Sports and Culture of Japan.

†Interfaculty Initiative in Information Studies, The University of Tokyo, Japan. mailto:matsuo@mist.i.u-tokyo.ac.jp

where $\dot{\boldsymbol{z}} = (d/dt)\boldsymbol{z}(t)$ and '$(\cdot)^{\mathrm{T}}$' denotes transpose of the matrix. Therefore, in this case, we call the system (1) "conservative." Hamiltonian systems, in which $N = 2M$ ($M \in \{1, 2, \dots\}$) and

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

is an example. On the other hand, when A is negative definite, H decreases along the solution:

$$\frac{\mathrm{d}}{\mathrm{d}t} H(\boldsymbol{z}(t)) = (\nabla H)^{\mathrm{T}} A \nabla H \leq 0, \tag{3}$$

and hence we call the system (1) "dissipative." Hamiltonian systems with a damping term, in which

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

is an example.

For the conservative or dissipative systems, it is widely accepted that numerical schemes that inherit the conservation or dissipation property, i.e., "dissipative" or "conservative" schemes, are desirable, and thus many efforts have been devoted in this theme[1]. In the '70s and '80s, associated with specific problems, several conservative or dissipative schemes of first- or second-order accuracy were presented (e.g. [2, 6]). Then, in the '90s, general methods for designing conservative or dissipative schemes of first- or second-order accuracy have been independently proposed by McLachlan *et al.*[14] (see also Gonzalez[5]), and Furihata and Matsuo([3, 11, 12]). 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.

In contrast to these successes, relatively little had been known on the third- or higher-order schemes. For a long period, the only tool had been the so-called composition technique[15], by which we can construct high-order conservative schemes. The first work which dealt with both dissipative and conservative cases in a complete manner was Matsuo[13], where a method was proposed for automatic designing of high-order dissipative or conservative schemes. The method had an additional advantage that the computational cost of the resulting schemes are relatively very cheap. But unfortunately the method had one weak point: the resulting schemes can suffer

[1]For the conservative systems, there are also *nearly* conservative methods, such as the symplectic methods for the Hamiltonian systems, and some symmetric methods for reversible systems[7, 8]. In this paper, however, we are mainly concerned with *strictly* conservative and dissipative methods.

the same instability as the so-called "backward difference formula"(BDF) schemes. Thus, from the practical point of view, the attainable order was limited to the 6th as in BDF.

In this paper, we propose a new method which overcomes the order barrier. The key point is to utilize the idea of the "*generalized* BDF" (GBDF)[1]. We also present an efficient "block" algorithm to reduce the computational cost. The rest of this paper is organized as follows. In Section 2, we briefly review the existing (2nd- and conventional higher-order) methods. In Section 3, we propose a new method for designing arbitrary high-order schemes, by utilizing the idea of GBDF. Section 4 is devoted to a "block" algorithm of the proposed schemes. Section 5 is for concluding remarks.

2 The existing methods

The existing methods up to the sixth order are briefly reviewed.

2.1 Second-order methods

The second-order method introduced in [3, 5, 11, 14] is summarized. The method is based on the concept of the "discrete derivative," or the "discrete gradient." Using this concept, schemes are defined analogously to the original system (1), so that the conservation or dissipation property results.

Let Δt be a time-mesh size. Numerical solutions are denoted by $\boldsymbol{z}^{(m)} \simeq \boldsymbol{z}(m\Delta t)$. In this section, $H(\boldsymbol{z}^{(m)})$ is often abbreviated as $H^{(m)}$. With these notations, the discrete derivative is defined as follows.

Definition 1 (First- or second-order discrete derivative) *For a sufficiently smooth function $f : \mathbf{R}^N \to \mathbf{R}$, we call $\nabla_{\mathrm{d}} f : \mathbf{R}^N \times \mathbf{R}^N \to \mathbf{R}^N$ "a discrete derivative" if it satisfies the following two conditions.*

1. *$f(\boldsymbol{y}_1) - f(\boldsymbol{y}_2) = (\nabla_{\mathrm{d}} f(\boldsymbol{y}_1, \boldsymbol{y}_2))^{\mathrm{T}} (\boldsymbol{y}_1 - \boldsymbol{y}_2)$ for all $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbf{R}^N$.*
2. *For any sufficiently smooth function $\boldsymbol{x}(t) : \mathbf{R} \to \mathbf{R}^N$, any $t_1, t_2 \in \mathbf{R}$, and p=1 or 2, there exist $\tilde{t}(t_1, t_2)$ such that $\nabla_{\mathrm{d}} f(\boldsymbol{x}(t_1), \boldsymbol{x}(t_2)) = \nabla f(\boldsymbol{x}(\tilde{t})) + O(|t_1 - t_2|^p)$ holds.*

Furthermore, we call $\nabla_{\mathrm{d}} f$ "a first-order discrete derivative at $\tilde{t}$" when $p = 1$, and a "second-order discrete derivative at $\tilde{t}$" when $p = 2$.

The above is a formal definition. There are several concrete examples that satisfy the definition[13]. We here present an example given by Gonzalez[5].

$$\nabla_{\mathrm{d}} f(\boldsymbol{x}, \boldsymbol{y}) = \nabla f(\boldsymbol{z}) + \frac{f(\boldsymbol{x}) - f(\boldsymbol{y}) - (\nabla f(\boldsymbol{z}))^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{y})}{\|\boldsymbol{x} - \boldsymbol{y}\|_2^2} (\boldsymbol{x} - \boldsymbol{y}), \quad \boldsymbol{z} = \frac{\boldsymbol{x} + \boldsymbol{y}}{2}, \tag{4}$$

where $\|\cdot\|_2$ is the Euclidean norm.

Once a concrete form of discrete derivative is given, a conservative or dissipative scheme can be constructed as follows.

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

$$\begin{cases} \frac{\boldsymbol{z}^{(m+1)}-\boldsymbol{z}^{(m)}}{\Delta t} = A\nabla_{\mathrm{d}} H(\boldsymbol{z}^{(m+1)}, \boldsymbol{z}^{(m)}) \quad (m = 1, 2, \dots), \\ \boldsymbol{z}^{(0)} = \boldsymbol{z}_0 \quad \text{(an initial value)} \end{cases} \tag{5}$$

is conservative:

$$H^{(m)} = H^{(0)} \quad (m = 1, 2, \dots), \tag{6}$$

if A is skew-symmetric, or is dissipative:

$$H^{(m+1)} \leq H^{(m)} \quad (m = 1, 2, \dots), \tag{7}$$

if A is negative definite. The scheme is second-order, if the discrete derivative $\nabla_{\mathrm{d}} H$ is second-order at $\tilde{t} = ((m+1)\Delta t + m\Delta t)/2$. Otherwise, the scheme is first-order.

2.2 High-order method

The high-order method proposed by Matsuo[13] is briefly reviewed. The key point is to replace the low (second) order difference operator in the second-order method with high-order ones to increase accuracy. To this end, pth-order difference operators are introduced first as follows.

Definition 2 (pth-order difference operator) *Let $\delta^{\langle 1 \rangle}_{m;\boldsymbol{c}}$ be a difference operator defined as*

$$\delta^{\langle 1 \rangle}_{m;\boldsymbol{c}} \boldsymbol{z}^{(m)} \equiv \sum_{i=-l_1}^{l_2} \frac{c_i \boldsymbol{z}^{(m+i)}}{\Delta t}, \tag{8}$$

where $l_1, l_2 \in \{0, 1, 2, \dots\}, c_i \in \mathbf{R}$, and $\boldsymbol{c} = (c_{-l_1}, \dots, c_{l_2})$. If for any sufficiently smooth function $\boldsymbol{x}(t) : \mathbf{R} \to \mathbf{R}^N$ and any $t \in \mathbf{R}$, there exists $\tilde{t}(t, \Delta t, l_1, l_2)$ such that

$$\sum_{i=-l_1}^{l_2} \frac{c_i \boldsymbol{x}(t + i\Delta t)}{\Delta t} = \dot{\boldsymbol{x}}(\tilde{t}) + O(\Delta t^p) \tag{9}$$

holds, we call $\delta^{\langle 1 \rangle}_{m;\boldsymbol{c}}$ "a pth-order difference operator at $\tilde{t}$" which refers $l = l_1 + l_2 + 1$ points, and denote it by $\delta^{\langle 1 \rangle, p}_{m;\boldsymbol{c}}$.

Usually, l is chosen to the smallest number: $l = p+1$. Next, a high-order discrete derivative is defined accordingly.

Definition 3 (pth-order discrete derivative) *Let $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}}$ a pth-order difference operator at $\tilde{t}$ which refers $l = l_1 + l_2 + 1$ points. For a sufficiently smooth function $f : \mathbf{R}^N \to \mathbf{R}$, we call*

$$\nabla^p_{\mathrm{d};\boldsymbol{c}} f : \underbrace{\mathbf{R}^N \times \cdots \times \mathbf{R}^N}_{l} \to \mathbf{R}^N$$

"a pth-order discrete derivative (with respect to $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}}$)," if the function satisfies the following conditions.

1. $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} f(\boldsymbol{y}_m) = \left(\nabla^p_{\mathrm{d};\boldsymbol{c}} f(\boldsymbol{y}_{m-l_1}, \dots, \boldsymbol{y}_{m+l_2})\right)^{\mathrm{T}} \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} \boldsymbol{y}_m$, *for any* $\boldsymbol{y}_{m-l_1}, \dots, \boldsymbol{y}_{m+l_2} \in \mathbf{R}^N$;
2. *For any sufficiently smooth function $\boldsymbol{x}(t) : \mathbf{R} \to \mathbf{R}^N$ and any $t \in \mathbf{R}$, $\nabla^p_{\mathrm{d};\boldsymbol{c}} f(\boldsymbol{x}(t - l_1\Delta t), \dots, \boldsymbol{x}(t + l_2\Delta t)) = \nabla f(\boldsymbol{x}(\tilde{t})) + O(\Delta t^p)$ holds.*

The above is a formal definition. A concrete example is given below as an extension of the Gonzalez's second-order discrete derivative (4).

Theorem 1 (A pth-order discrete derivative[13]) *Let $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}}$ be a pth-order difference operator at $\tilde{t}$ which refers $l = l_1 + l_2 + 1$ points. Then, the following is a pth-order discrete derivative.*

$$\nabla^p_{\mathrm{d};\boldsymbol{c}} f = \nabla f(\tilde{\boldsymbol{y}}) + \frac{\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} f(\boldsymbol{y}_m) - (\nabla f(\tilde{\boldsymbol{y}}))^{\mathrm{T}} \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} \boldsymbol{y}_m}{\|\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} \boldsymbol{y}_m\|_2^2} \, \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} \boldsymbol{y}_m,$$

where $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{y}}(\boldsymbol{y}_{m-l_1}, \dots, \boldsymbol{y}_{m+l_2})$ is a function of $\boldsymbol{y}_{m-l_1}, \dots, \boldsymbol{y}_{m+l_2} \in \mathbf{R}^N$ such that for any sufficiently smooth function $\boldsymbol{x}(t) : \mathbf{R} \to \mathbf{R}^N$ and any $t \in \mathbf{R}$, $\tilde{\boldsymbol{y}}(\boldsymbol{x}(t - l_1\Delta t), \dots, \boldsymbol{x}(t + l_2\Delta t)) = \boldsymbol{x}(\tilde{t}) + O(\Delta t^p)$ holds.

The discrete derivative consists of two terms: the first term is the "true" derivative at $\tilde{y}$; and the second term is the pth-order correction term which enforces the discrete chain rule (and thus essential in conservative/dissipative schemes). Using the discrete derivative, a pth-order conservative or dissipative scheme for (1) can be constructed as follows.

Scheme 2 (pth-order scheme) *Let $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}}$ be a pth-order difference operator at $\tilde{t}$ which refers $l = l_1 + l_2 + 1$ points, and let $\nabla^p_{\mathrm{d}} H$ be the pth-order discrete derivative of H at $\tilde{t}$. Then, a scheme*

$$\begin{cases} \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} \boldsymbol{z}^{(m)} = A \nabla^p_{\mathrm{d};\boldsymbol{c}} H(\boldsymbol{z}^{(m-l_1)}, \dots, \boldsymbol{z}^{(m+l_2)}) \quad (m = l_1, l_1 + 1, \dots), \\ \boldsymbol{z}^{(0)} = \boldsymbol{z}_0 \quad \text{(an initial-value)}, \\ \boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(l-2)} \quad \text{(starting-values)}, \end{cases} \tag{10}$$

is conservative:

$$\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} H^{(m)} = 0 \quad (m = l_1, l_1 + 1, \dots), \quad \text{if } A \text{ is skew-symmetric}, \tag{11}$$

or is dissipative:

$$\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}} H^{(m)} \le 0 \quad (m = l_1, l_1+1, \dots), \quad \textit{if } A \textit{ is negative definite.} \tag{12}$$

In addition, the scheme is pth-order.

Scheme 2 is not self-starting. We have to provide the starting-values $\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(l-2)}$ in addition to the initial value $\boldsymbol{z}^{(0)} = \boldsymbol{z}_0$. Usually, they are computed using another scheme with sufficient accuracy, such as Scheme 1 or the Runge-Kutta methods.

Note that the above story holds good, at least formally, for any pth-order difference operator $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}}$. But for the stability of the resulting scheme, we are forced to choose the operator to the so-called "stiffly-stable" backward difference operator: that is, the scheme should be like the BDF (Backward Difference Formula) scheme[9]. The usual linear stability analysis shows that BDF schemes are stable only up to the 6th order. Thus, unfortunately Scheme 2 is of practical use only in the 6th- or lesser order situations.

3 The new method: a GBDF approach

To overcome the order barrier, the idea of the "Generalized BDF" (GBDF) proposed by Brugunano and Trigiante[1] is utilized. The idea of GBDF is to solve a finite difference scheme not sequentially from start to end (i.e., as a initial value method), but simultaneously as a big system of equations (i.e., as a boundary value method). Brugunano and Trigiante showed that in this approach BDF-like schemes which are stable up to arbitrary order can be constructed (see [1] for the detail).

Let us consider the problem in $0 \le t \le T$ and its pth-order numerical solution. Let ν be a constant such that

$$\nu = \frac{p+2}{2}, \text{ if } p \text{ is even}; \quad \nu = \frac{p+1}{2}, \text{ otherwise.} \tag{13}$$

For a given large integer N_t, $N_t + p - \nu - 1$ time mesh points are set, i.e., $\Delta t = T/(N_t + p - \nu - 1)$. The numerical solutions $\boldsymbol{z}^{(0)}, \dots, \boldsymbol{z}^{(N_t+p-\nu-1)}$ are then divided into three groups: The initial group $\boldsymbol{z}^{(0)}, \dots, \boldsymbol{z}^{(\nu-1)}$, the main group $\boldsymbol{z}^{(\nu)} \dots, \boldsymbol{z}^{(N_t-1)}$, and the final group $\boldsymbol{z}^{(N_t)}, \dots, \boldsymbol{z}^{(N_t+p-\nu-1)}$.

In each group, different versions of Scheme 2 are applied. More specifically, different time difference operators $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$ are used depending on the time step m. The operators are defined as follows.

Definition 4 (Special pth-order difference operator) *Let l_1, l_2 be the values defined in Table 1 for each time step $m \ge 0$. We define $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$ to be special versions of the pth-order difference operators $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}}$, whose coefficients $\boldsymbol{c}_m$ are uniquely determined, for each m, so that the operator be pth-order at $\tilde{t} = m\Delta t$.*

Table 1: Parameters in $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$

group	(l_1, l_2)	referred points
initial	$(m, p-m)$	$0, \dots, p$
main	$(\nu, p-\nu)$	$m-\nu, \dots, m+p-\nu$
final	$(m-N_t, N_t-m+p)$	$N_t-\nu-1, \dots, N_t+p-\nu-1$

The way of finding such coefficients can be found, for example, in [1]. The seventh-order case will be shown in Example 1. In Table 1, the referred points by each difference operators are also listed for readers' convenience. Notice that in the initial and final groups the referred points are fixed and l_1, l_2 vary depending on m (to avoid referring outside points), while on the other hand l_1, l_2 are fixed in the main group.

With the difference operator, now the *generalized* version of the pth-order scheme is defined as follows.

Scheme 3 (Generalized pth-order scheme) *Let $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$ be the difference operator defined as above, and $\nabla^p_{\mathrm{d};\boldsymbol{c}_m}$ be the pth-order discrete derivative defined in Theorem 1 as to $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$. Then, a scheme*

$$\begin{cases} \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}\boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m}H(\boldsymbol{z}^{(0)}, \dots, \boldsymbol{z}^{(p)}) & (m = 1, \dots, \nu-1), \\ \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}\boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m}H(\boldsymbol{z}^{(m-\nu)}, \dots, \boldsymbol{z}^{(m+p-\nu)}) & (m = \nu, \dots, N_t-1), \\ \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}\boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m}H(\boldsymbol{z}^{(N_t-\nu-1)}, \dots, \boldsymbol{z}^{(N_t+p-\nu-1)}) & (m = N_t, \dots, N_t+p-\nu-1), \\ \boldsymbol{z}^{(0)} = \boldsymbol{z}(0) & (\text{an initial value}), \end{cases} \tag{14}$$

is conservative:

$$\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}H^{(m)} = 0 \quad (m = l_1, l_1+1, \dots), \quad \textit{if } A \textit{ is skew-symmetric}, \tag{15}$$

or is dissipative:

$$\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}H^{(m)} \le 0 \quad (m = l_1, l_1+1, \dots), \quad \textit{if } A \textit{ is negative definite}. \tag{16}$$

In addition, the scheme is pth-order.

(Proof) We here abbreviate $H(\boldsymbol{z}^{(m-\nu)}, \dots, \boldsymbol{z}^{(m+p-\nu)})$ (or the similar terms in initial and final parts) as $H^{(m)}$. The special pth-order difference operator $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$ satisfies, as the special case of Theorem 1, the discrete chain rule

$$\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}H^{(m)} = \left(\nabla^p_{\mathrm{d};\boldsymbol{c}_m}H^{(m)}\right)^{\mathrm{T}}\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}\boldsymbol{z}^{(m)}.$$

This, together with the definition of the scheme, shows

$$\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m} H^{(m)} = \left(\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H^{(m)}\right)^{\mathrm{T}} A \left(\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H^{(m)}\right),$$

from which the conservation property (15) and the dissipation property (16) are clear. The claim on the order is trivial since both sides of the scheme is everywhere pth-order. ∎

To help readers' understanding, the 7th-order case is demonstrated below.

Example 1 (The 7th-order scheme) To implement Scheme 3, we have to specify the coefficients of the difference operators $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$ used in the scheme. (Remember that the discrete derivative $\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H$ also depends on the forms of the difference operators $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$. See Theorem 1.) There are eight different 7th-order operators, all of which refer $l = 7 + 1 = 8$ points[2]. The coefficients are listed in Table 2, being categorized according to the time step m. The bold face values denote that, the corresponding operators are 7th-order at those points; for example, for $m = 1$, the operator is 7th-order at $t = \Delta t$. Because now $\nu = (7 + 1)/2 = 4$ in (13), the first four solutions $\boldsymbol{z}^{(0)}, \ldots, \boldsymbol{z}^{(3)}$ belong to the initial group, and three sets of coefficients are used to construct three different operators at $m = 1, 2, 3$, needed in the first line of (14). Throughout the main group ($m = 4, \ldots, N_t - 1$), one operator is constantly used. Finally, since $N_t + 7 - 4 - 1 = N_t + 2$, three solutions $\boldsymbol{z}^{(N_t)}, \boldsymbol{z}^{(N_t+1)}, \boldsymbol{z}^{(N_t+2)}$ belong to the final group, and the three sets of coefficients are used for them.

Scheme 3 is expected to be very stable up to arbitrary order, inheriting the strong stability of the standard GBDF. As noted earlier, the discrete derivative $\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H$ defined in Theorem 1 is a combination of the "original" gradient ∇H and the pth-order correction term. If the correction term is switched off, Scheme 3 is nothing but the standard GBDF scheme whose linear stability is well established.

We have to pay, however, the price of this extra stability. Scheme 3 apparently involves a big $(N \times (N_t + p - \nu - 1))$-dimensional system of nonlinear equations, which must be solved iteratively by, for example, the Newton method. This requires not only heavy computation, but the complicated implementation task. As for the computational cost, an efficient algorithm will be presented in Section 4. To decrease the implementation complexity, some reliable Newton procedure library which automatically generates numerical Jacobian can be used.

[2]The first set of coefficients, i.e. the set for $m = 0$, is not used here since we do not need the equation for the given initial value $\boldsymbol{z}^{(0)} = \boldsymbol{z}_0$ in Scheme 3. The set will be used in the next section (Scheme 4).

Table 2: The coefficients of the 7th-order operators

time step (m)	c_{-l_1}	c_{-l_1+1}	$\cdots$			$\cdots$	c_{l_2-1}	c_{l_2}
0	$-\frac{\mathbf{363}}{\mathbf{140}}$	7	$-\frac{21}{2}$	$\frac{35}{3}$	$-\frac{35}{4}$	$\frac{21}{5}$	$-\frac{7}{6}$	$\frac{1}{7}$
1	$-\frac{1}{7}$	$-\frac{\mathbf{29}}{\mathbf{20}}$	3	$-\frac{5}{2}$	$\frac{5}{3}$	$-\frac{3}{4}$	$\frac{1}{5}$	$-\frac{1}{42}$
2	$\frac{1}{42}$	$-\frac{1}{3}$	$-\frac{\mathbf{47}}{\mathbf{60}}$	$\frac{5}{3}$	$-\frac{5}{6}$	$\frac{1}{3}$	$-\frac{1}{12}$	$\frac{1}{105}$
3	$-\frac{1}{105}$	$\frac{1}{10}$	$-\frac{3}{5}$	$-\frac{\mathbf{1}}{\mathbf{4}}$	1	$-\frac{3}{10}$	$\frac{1}{15}$	$-\frac{1}{140}$
$4, \ldots, N_t-1$	$\frac{1}{140}$	$-\frac{1}{15}$	$\frac{3}{10}$	-1	$\frac{\mathbf{1}}{\mathbf{4}}$	$\frac{3}{5}$	$-\frac{1}{10}$	$\frac{1}{105}$
N_t	$-\frac{1}{105}$	$\frac{1}{12}$	$-\frac{1}{3}$	$\frac{5}{6}$	$-\frac{5}{3}$	$\frac{\mathbf{47}}{\mathbf{60}}$	$\frac{1}{3}$	$-\frac{1}{42}$
N_t+1	$\frac{1}{42}$	$-\frac{1}{5}$	$\frac{3}{4}$	$-\frac{5}{3}$	$\frac{5}{2}$	-3	$\frac{\mathbf{29}}{\mathbf{20}}$	$\frac{1}{7}$
N_t+2	$-\frac{1}{7}$	$\frac{7}{6}$	$-\frac{21}{5}$	$\frac{35}{4}$	$-\frac{35}{3}$	$\frac{21}{2}$	-7	$\frac{\mathbf{363}}{\mathbf{140}}$

Here we present three numerical examples. In all the examples in this paper, the routine NEQNF included in IMSL Fortran library, which is a Newton solver with numerical Jacobian, is used.

Example 2 First, the simplest equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}z(t) = -\nabla H(z), \quad H(z) = \frac{z^2}{2}, \tag{17}$$

is considered in the interval $0 \le t \le 10$, to check the accuracy of Scheme 3. As one can easily check, it is dissipative: $\frac{\mathrm{d}}{\mathrm{d}t}H(z(t)) \le 0$.

The problem is solved by three schemes, GBDFn(d)($n = 7, 9, 11$), which are the 7th-, 9th-, and 11th-order versions of Scheme 3, respectively. The suffix "(d)" denotes that the scheme is dissipative, i.e., not the standard GBDF's. Fig. 1 shows the error in the numerical solutions at $t = 10$. The lines are the guide lines which show 7th-, 9th-, and 11th-order, respectively. All of three schemes run quite stably and are rightly 7th-, 9th-, and 11th-order. Fig. 2 shows the numerical solution (points) against the exact solution (solid line). The dissipation property is clear.

Example 3 (Harmonic oscillator) The harmonic oscillator:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{z}(t) = J\nabla H(\boldsymbol{z}), \quad \boldsymbol{z} = (z_1, z_2)^{\mathrm{T}}, \ H = \omega\left(\frac{z_1^2 + z_2^2}{2}\right), \tag{18}$$

where

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

and $\omega \in \mathbf{R}$ is the angular velocity, is considered. We took $\omega = 3/2$ and integrated the problem in the interval $0 \le t \le 100$, with the time mesh size

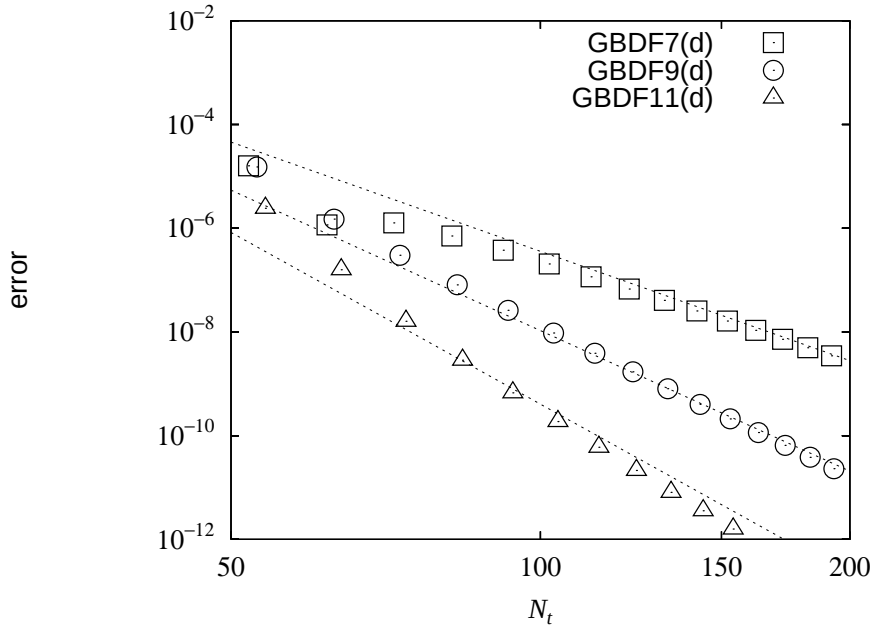

Figure 1: [Example 2] Errors in the 7th-, 9th-, and 11th-order schemes

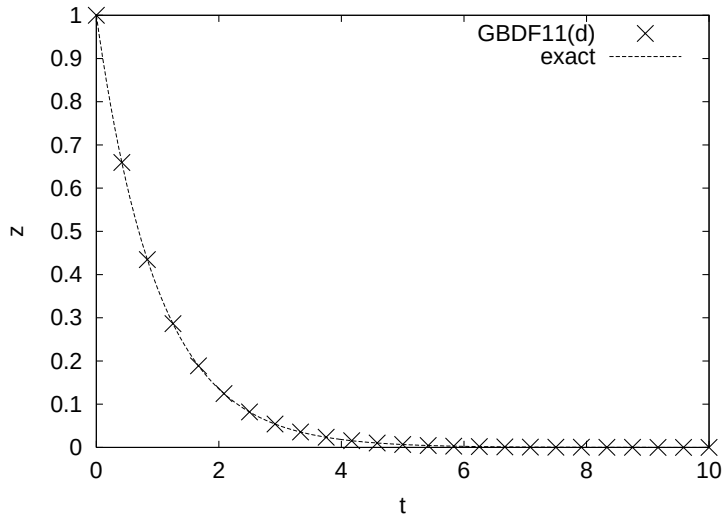

Figure 2: [Example 2] The numerical and exact solution (the 11th-order case, $N_t = 20$)

$\Delta t = 1/2$ (this means the numerical solution has about 8 points in each oscillation cycle). The problem is solved by three schemes:

BDF7 — The standard 7th-order BDF scheme, which is unstable.

GBDF7 — The 7th-order GBDF scheme, as is proposed by Brugunano and Trigiante[1]. This scheme is not conservative.

GBDF7(c) — Scheme 3 of order 7, which is conservative.

Note that the harmonic oscillator problem is a *linear* problem, for which the standard GBDF schemes are proved to work quite stably (see Brugunano and Trigiante[1]). In GBDF7 and GBDF7(c), the dimension of the nonlinear system is $2 \times (200 + 7 - 4 - 1) = 404$.

Fig. 3 shows the trajectories of the numerical and exact solutions. BDF7 blows up as expected since it is an unstable scheme. GBDF7 is quite stable as expected, but its solutions do not stay exactly on the true orbit (circle). GBDF7(c) is stable, and its solutions exactly stay on the true orbit.

Fig. 4 shows the evolutions of the energies. The energy by BDF7 rapidly blows up. The energy by GBDF7 slowly decreases toward zero, which reflects the fact that the scheme is stable (and not conservative). The energy by GBDF7(c) is strictly conserved as expected.

These results illustrate that GBDF schemes are in fact quite stable, for the linear problem, and in addition to that the conservation of energy further refines the result.

Example 4 (Kepler problem) The kepler problem:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{z}(t) = J\nabla H(\boldsymbol{z}), \quad H(\boldsymbol{z}) = \frac{z_1^2 + z_2^2}{2} - \frac{1}{\sqrt{z_3^2 + z_4^2}}, \tag{19}$$

where

$$J = \begin{pmatrix} 0 & -I_2 \\ I_2 & 0 \end{pmatrix}, \quad I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

is considered. Note that the Kepler problem is a *nonlinear* problem, where the stability of the standard GBDF schemes is not well established. In all of the following experiments, we took the initial data $\boldsymbol{z}(0) = (0, 3, 0.2, 0)$ (which corresponds to the Kepler orbit of eccentricity 0.8, and cycle 2π), and tested the schemes GBDF7 and GBDF7(c).

Firstly, let us focus on the qualitative aspects of the schemes. To this end, the problem is integrated in the interval $0 \le t \le 10$ with time mesh size $\Delta t = 1/50$. In these settings, the dimension of the nonlinear system is $4 \times (500 + 7 - 4 - 1) = 2008$.

Fig. 5 shows the trajectories. Both schemes run quite stably, and we can not see any significant difference between GBDF7 and GBDF7(c). This

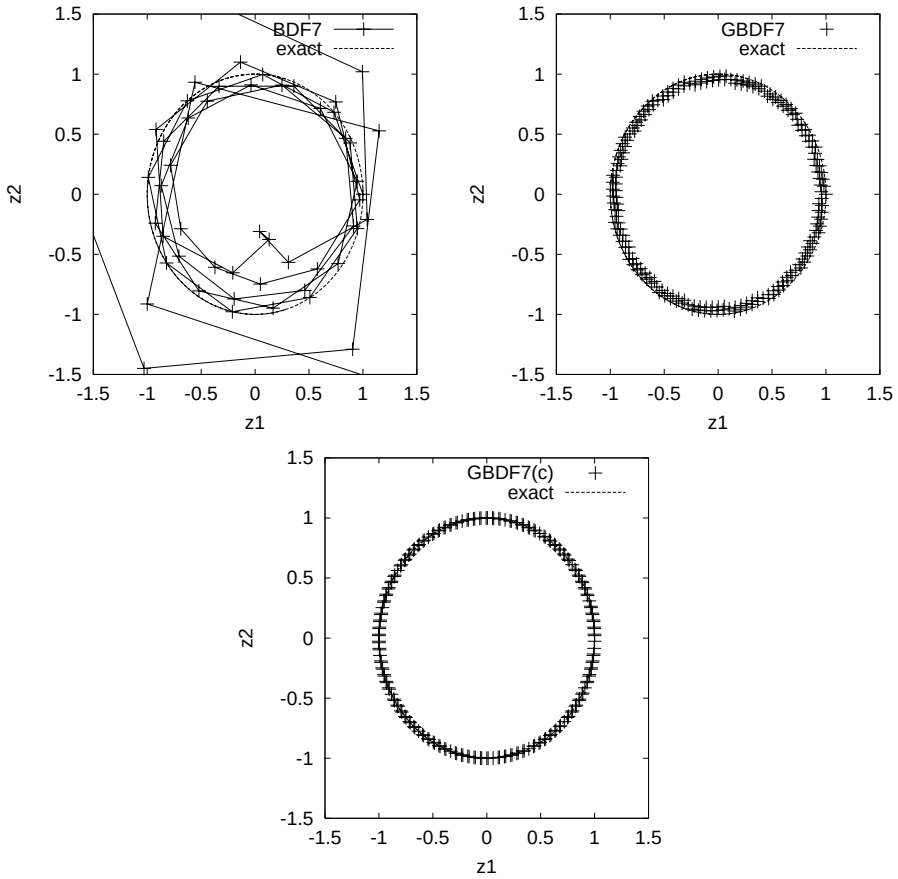

Figure 3: [Example 3] The trajectories of the numerical and exact solutions in Example 2: (top left) BDF7, (top right) GBDF7, (bottom) GBDF7(c)

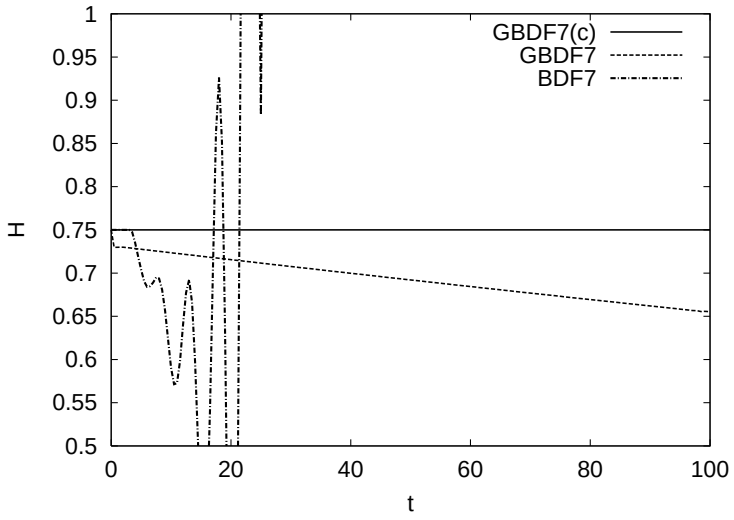

Figure 4: [Example 3] The evolutions of the energies

should be attributed to the fact that essentially both schemes are quite accurate thanks to the high-order (7th-order) accuracy.

On the other hand, in Fig. 6, which shows the evolutions of the energies, the difference is rather clear. In GBDF7(c) the energy is strictly conserved, while in GBDF7 it is slowly dissipated, like as in the harmonic oscillator problem. But this time, the energy by GBDF7 shows undesirable oscillations at perihelion ($t \sim 2\pi$). Thus in this nonlinear problem, we can conclude that the conservation of energy is desired for stabilizing the numerical solution.

Secondly, we turn our attention to the computational costs of the schemes. The problem is integrated in the interval $0 \leq t \leq 1$, with several different Δt (or N_t). Table 3 shows the computation times of GBDF7 and GBDF7(c) under several N_t. The experiment was done using a PC with Pentium4 (2.8GHz), 1GB RAM, Windows XP, and Intel Fortran Compiler 8.0. In the table, the columns named "(growth)" show the ratio $\log_{N_2/N_1}(T_2/T_1)$ (T_1, T_2 are the computation times required when $N_t = N_1$ and N_2, respectively), which gives the constant c for $O({N_t}^c)$ quantities. We can see that roughly the computational costs are $O({N_t}^3)$. It is natural since in Scheme 3 roughly NN_t-dimensional system of nonlinear equations must be solved. Thus Scheme 3 might be too expensive when N_t is quite large. We also notice that GBDF7(c) requires slightly much costs since it involves extra calculations in evaluating the high-order discrete derivative $\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H^{(m)}$.

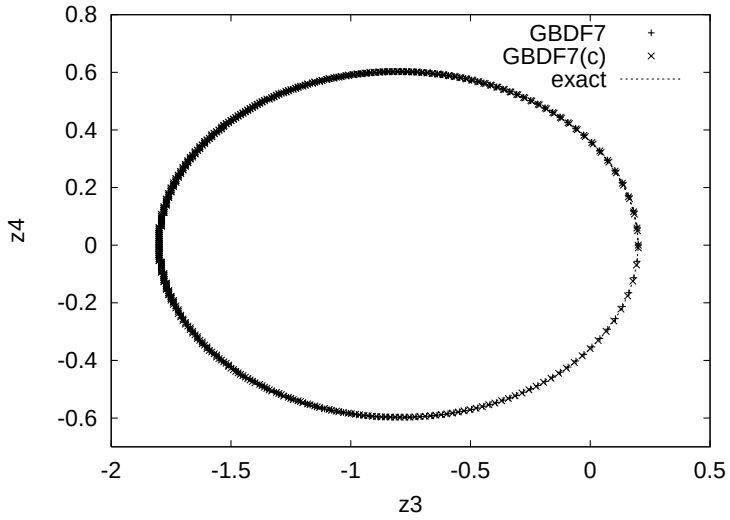

Figure 5: [Example 4] The trajectories of numerical and exact solutions

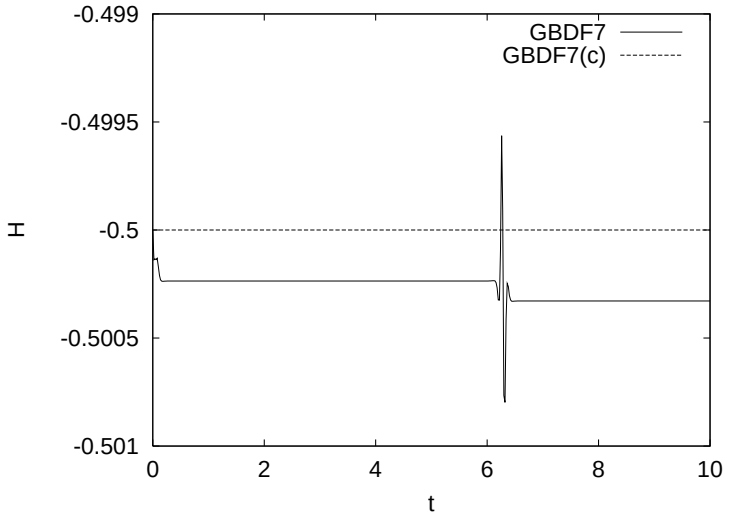

Figure 6: [Example 4] The evolutions of the energies

Table 3: [Example 4] Computation time (in seconds)

N_t	GBDF7	(growth)	GBDF7(c)	(growth)
200	2.7	—	3.1	—
300	8.7	2.9	9.8	2.8
400	19	2.7	22	2.8
500	37	3.0	42	2.9
600	63	2.9	72	3.0

4 A block algorithm

As pointed out in the previous section, Scheme 3 involves $O({N_t}^3)$ computation time, and thus may not be very practical as it stands. In this section a "block" algorithm is presented to overcome this difficulty. The idea here is to utilize Iavernaro and Mazzia's technique for general boundary value methods[10]. Hereafter we limit ourselves to the odd order schemes for simplicity. The even order cases can be treated similarly.

The smallest N_t for which Scheme 3 makes sense is $N_t = \nu + 1$. If we also treat the initial value ($m = 0$) as the unknown value, Scheme 3 with $N_t = \nu + 1$ becomes

$$\begin{cases} \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m} \boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H(\boldsymbol{z}^{(0)},\dots,\boldsymbol{z}^{(p)}) & (m = 0,\dots,\nu-1), \\ \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m} \boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H(\boldsymbol{z}^{(0)},\dots,\boldsymbol{z}^{(p)}) & (m = \nu), \\ \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m} \boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H(\boldsymbol{z}^{(0)},\dots,\boldsymbol{z}^{(p)}) & (m = \nu+1,\dots,p), \end{cases} \tag{20}$$

which consists of $(p+1)$ equations for $(p+1)$ unknown $\boldsymbol{z}^{(m)}$'s. Our strategy here is to repeatedly use this "smallest block" of the scheme to reach the "goal" time T. For example, if we are given the initial value $\boldsymbol{z}^{(0)}$, we solve (20) with it to obtain $\boldsymbol{z}^{(1)},\dots,\boldsymbol{z}^{(p)}$. Then we repeat this process with the new initial value $\boldsymbol{z}^{(p)}$, and so on, until $sp\Delta t \geq T$ (s is the number of cycles). In each cycle, we only have to solve $(N \times (p+1))$-dimensional system of nonlinear equations (which is the smallest). The overall computational cost would be, roughly speaking, proportional to the number of cycles (i.e. $O(N_t)$), which is far smaller than $O({N_t}^3)$ in the previous section.

But as it is shown in simpler settings[10], this "naive" strategy does not work as it stands, since the strong stability of GBDF would be lost by splitting the interval. Thus, based on the clever strategy in [10], we here propose a slightly different algorithm.

Scheme 4 (Block version of the pth-order scheme) *Let $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$ be the difference operator defined as above, and $\nabla^p_{\mathrm{d};\boldsymbol{c}_m}$ is the pth-order discrete derivative defined in Theorem 1 as to $\delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m}$. Assume that the initial and*

starting values at $m = 0, 1, \ldots, \nu-1$ *are given. Then we calculate* $\boldsymbol{z}^{(\nu)}, \boldsymbol{z}^{(\nu+1)}, \ldots$ *using the algorithm:*

(a) $s := \nu$*;*

(b) With the known values $\boldsymbol{z}^{(s-\nu)}, \ldots, \boldsymbol{z}^{(s-1)}$*, solve*

$$\begin{cases} \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m} \boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H(\boldsymbol{z}^{(s-\nu)}, \ldots, \boldsymbol{z}^{(s+p-\nu)}) & (m = s-\nu, \ldots, s-1), \\ \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m} \boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H(\boldsymbol{z}^{(s-\nu)}, \ldots, \boldsymbol{z}^{(s+p-\nu)}) & (m = s), \\ \delta^{\langle 1\rangle,p}_{m;\boldsymbol{c}_m} \boldsymbol{z}^{(m)} = A\nabla^p_{\mathrm{d};\boldsymbol{c}_m} H(\boldsymbol{z}^{(s-\nu)}, \ldots, \boldsymbol{z}^{(s+p-\nu)}) & (m = s+1, \ldots, s+p-\nu), \end{cases} \tag{21}$$

to obtain $\boldsymbol{z}^{(s)}, \ldots, \boldsymbol{z}^{(s+p-\nu)}$*;*

(c) Discard $\boldsymbol{z}^{(s+1)}, \ldots, \boldsymbol{z}^{(s+p-\nu)}$*; let* $s := s+1$*; then go to (b) until* $s = N_t$*.*

In the above algorithm, we still solve $(N \times (p+1))$-dimensional system of nonlinear equations in each cycle. This time, however, only one numerical solution $\boldsymbol{z}^{(s)}$ is adopted in each cycle, and the rests $(\boldsymbol{z}^{(s+1)}, \ldots, \boldsymbol{z}^{(s+p-\nu)})$ are discarded. It may seem an waste, but it greatly helps the stability of the overall scheme (see [10] for rigorous linear stability analysis of the standard block GBDF schemes). The starting values must be provided by, for example, another (conservative/dissipative) numerical scheme.

We show three numerical examples of Scheme 4.

Example 5 The simplest problem in Example 2 is considered again to check the accuracy of the schemes. The problem is solved by three schemes, BGBDFn(d)$(n = 7, 9, 11)$, which are the 7th-, 9th-, and 11th-order versions of Scheme 4, respectively. Fig. 7 shows the error in the numerical solutions at $t = 10$. All of three schemes run quite stably and are rightly of 7th-, 9th-, and 11th-order. Notice that the vertical scale is different from Fig. 1 in Example 2; Scheme 4 are more accurate than Scheme 3.

In what follows, Scheme 4 of 7th-order and the corresponding Iavernaro and Mazzia's 7th-order block GBDF (which is not conservative/dissipative) are considered. Scheme 4 is denoted as "BGBDF7(c)," and the corresponding block GBDF as "BGBDF7."

Example 6 (Kepler problem) The same problem as in Example 4 is considered. This is a conservative problem.

Firstly, the problem is integrated in the interval $0 \le t \le 500$ (which corresponds to about 80 periods), with the time mesh size $\Delta t = 1/40$ (thus the number of mesh points are $N_t = 20000$) to see its long time behaviour. Note that this setting is almost impossible for Scheme 3. If we applied Scheme 3 with these settings, we would have to solve 80,012-dimensional system of nonlinear equations. Fig. 8 and Fig. 9 shows the trajectories of

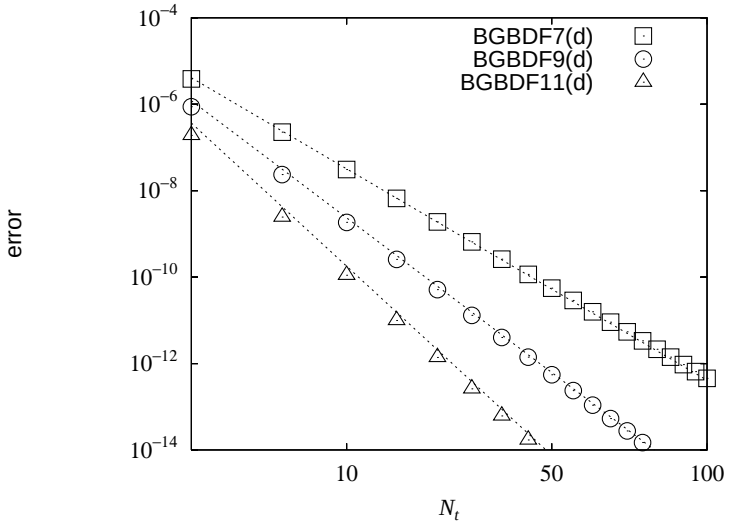

Figure 7: [Example 5] Errors in numerical solutions

BGBDF7 and BGBDF7(c), respectively. BGBDF7 (Fig. 8) stably calculates the numerical solution, but the orbit gradually falls into the center due to numerical dissipation. BGBDF7(c) (Fig. 9) gives better result, where the orbit never shrink.

Fig. 10 shows the energies in both schemes, which is more illustrative and impressive. The energy in BGBDF7(c) is rightly conserved. It is actually conserved to the machine accuracy. Thus we can see the conservation property holds good also in Scheme 4. In contrast to that, the energy in BGBDF7 gradually dissipates, and it also exhibits oscillations at the perihelion(s). Since we are now considering about 80 periods of the Kepler motion, we observe about 80 spikes in the graph, which agrees with the result in the previous section. We conclude that the qualitative aspects of the original schemes are inherited to the blocked version Scheme 4.

Secondly, we consider the same setting as in the second part of Example 4, to check the computation time. Table 4 shows the computation times of BGBDF7 and BGBDF7(c). Compare it with Table 3. At a glance, we understand that the computational cost is dramatically decreased; nearly 1/300 when $N_t = 600$. The growth order is much smaller than three. We expected it to be one, but the result is blurring. The reason of this should be attributed to the fact that the cost in each cycle varies and depends on the initial values set for the Newton iteration. It can be better (or even worse) by tuning the initial values. We here do not get involved with this issue any further, and leave it to the future work.

Thirdly, Fig. 11 shows the accuracy of the numerical solutions. The points are the accuracies of numerical solutions, and the dashed line is the

guide line to show $O(\Delta t^7)$. The solutions are rightly 7th-order. GBDF7(c) is slightly better thanks to the conservation property.

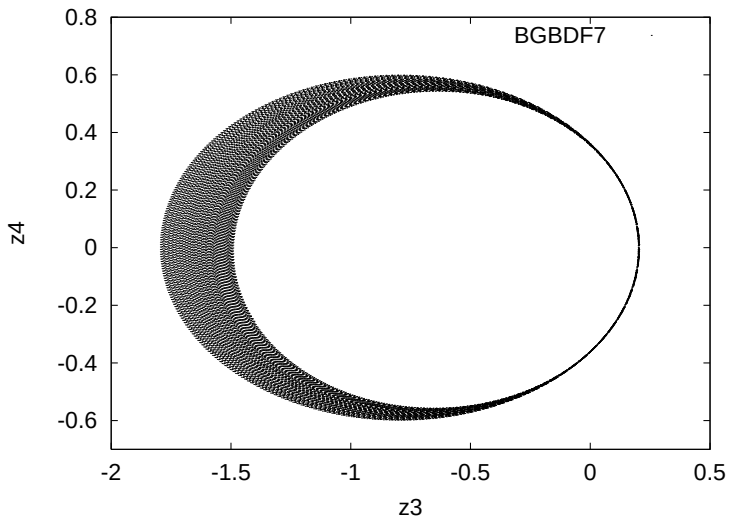

Figure 8: [Example 6] The trajectories of numerical solution (BGBDF7)

Table 4: [Example 6] Computation time (in seconds)

N_t	GBDF7	(growth)	GBDF7(c)	(growth)
200	0.074	—	0.078	—
300	0.099	0.72	0.112	0.89
400	0.128	0.89	0.145	0.90
500	0.176	1.43	0.189	1.19
600	0.230	1.47	0.228	1.03

Example 7 (Damped Kepler problem) A damped Kepler problem:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{z}(t) = J\nabla H(\boldsymbol{z}), \quad H(\boldsymbol{z}) = \frac{z_1^2 + z_2^2}{2} - \frac{1}{\sqrt{z_3^2 + z_4^2}}, \tag{22}$$

where

$$J = \begin{pmatrix} -\alpha I_2 & -I_2 \\ I_2 & 0 \end{pmatrix}, \quad I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

is considered. The constant $\alpha > 0$ is the resistance coefficient, and is taken to 0.001 in this experiment. The other parameters are the same as in Example 4

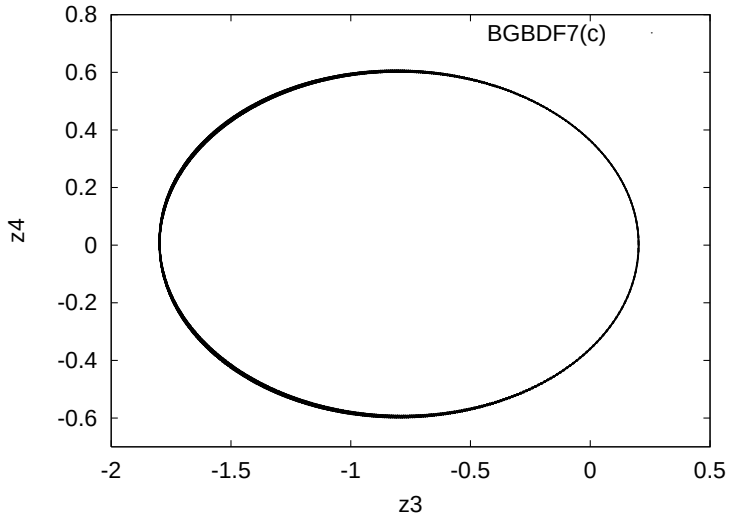

Figure 9: [Example 6] The trajectories of numerical solution (BGBDF7(c))

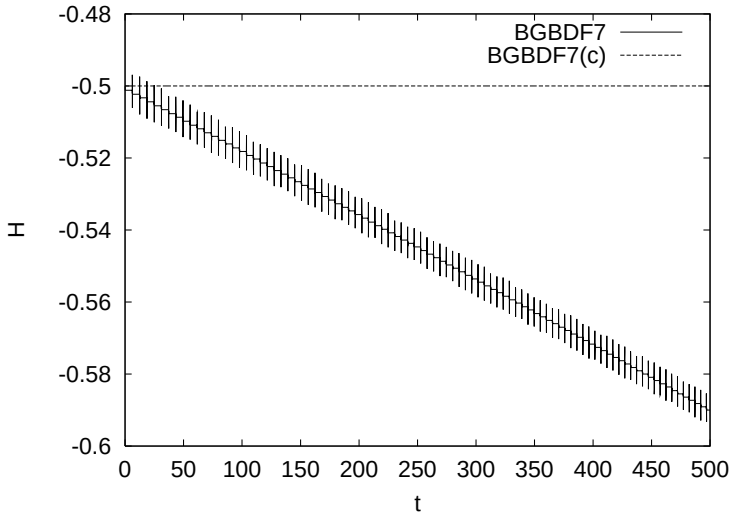

Figure 10: [Example 6] The evolutions of the energies

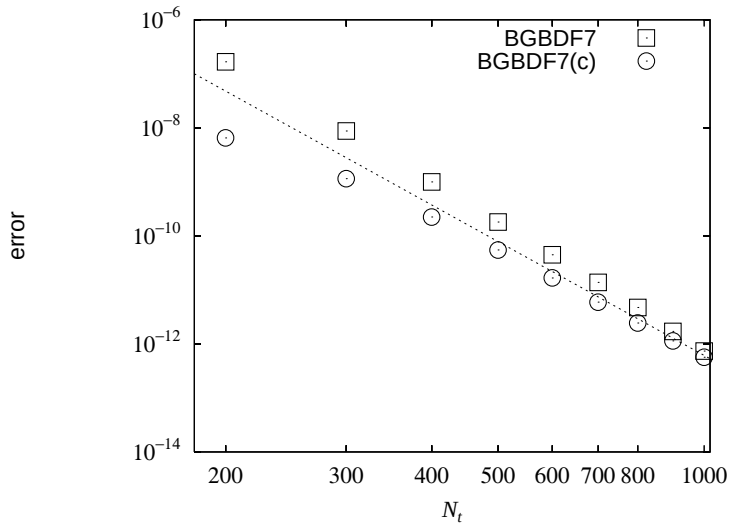

Figure 11: [Example 6] The accuracy of numerical solutions

and 6. Two schemes are compared: the dissipative Scheme 4 of 7th-order (denoted as BGBDF7(d)), and the corresponding Iavernaro and Mazzia's block GBDF7 (BGBDF7) which is not dissipative.

Fig. 12 and Fig. 13 shows the trajectories of numerical solutions by BGBDF7 and BGBDF7(d). In both graphs the trajectories falls into the center, but the solution by BGBDF7 falls faster than BGBDF7(d).

Fig. 14 shows the evolutions of the energies in both schemes. We can observe two facts: first, BGBDF7 shows oscillations at the perihelion(s), as in the conservative case (Example 6), while BGBDF7(d) strictly dissipates the energy. Thus we can see the dissipation property holds good also in Scheme 4. Second, the dissipation profile of BGBDF7(d) agrees quite well with the exact profile (which was obtained by a numerical calculation choosing very small time mesh size). In contrast to that, the dissipation in BGBDF7 is much larger than the original profile. This explains the difference between Fig. 12 and Fig. 13.

5 Concluding remarks

In this paper a new method for designing dissipative or conservative schemes of arbitrary high-order is proposed. In the method, the previously proposed method[13] is incorporated with the idea of the generalized backward difference formula. The order and stability of the proposed scheme are confirmed by numerical experiments. In particular, it is illustrated that the proposed schemes are superior to the standard GBDF schemes in the sense that they

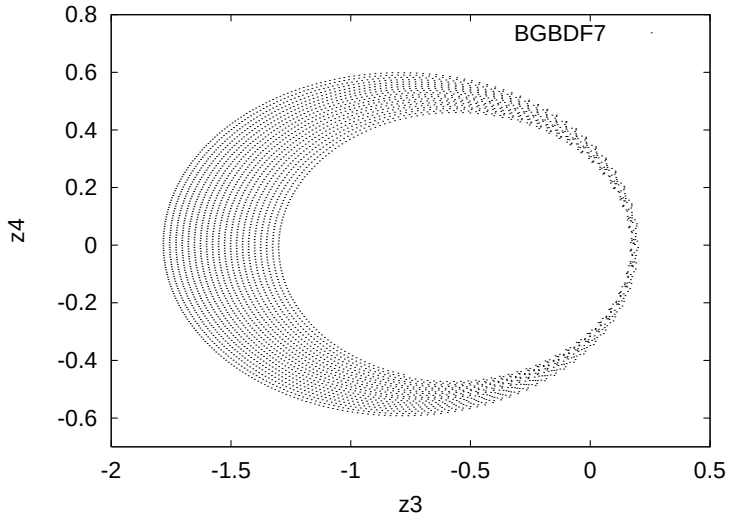

Figure 12: [Example 7] The trajectories of numerical solution (BGBDF7)

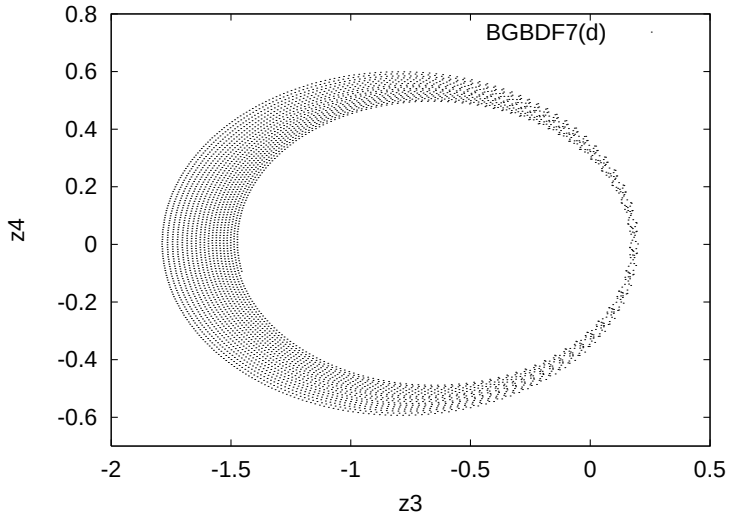

Figure 13: [Example 7]The trajectories of numerical solution (BGBDF7(d))

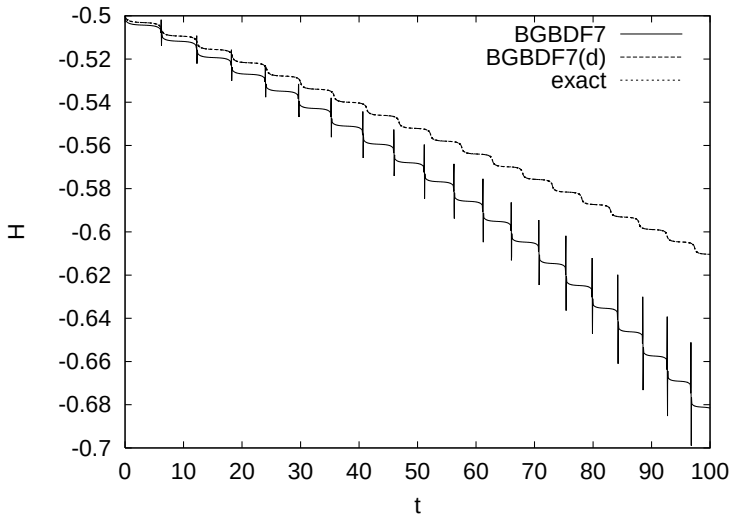

Figure 14: [Example 7]The evolutions of the energies

prohibit undesirable numerical oscillations with a little additional computational cost in some nonlinear problem. An efficient block algorithm, which dramatically saves the computational cost, was also presented.

There remains several issues as the future works. First, theoretical aspects of the proposed schemes, including the rigorous stability analysis, should be clarified. The standard tools for stability analysis, however, can not be simply applied to our case in the following two senses: firstly, the proposed schemes are *always* nonlinear even for linear problems; and secondly, considering Dahlquist's test equation does not make sense when we are concerned with conservative schemes. Some new concept and tools of stability analysis should be introduced to handle our case.

Second, it must be examined if the proposed schemes can be implemented with variable time mesh sizes, which is quite crucial in practical computation. We believe that the known techniques as to GBDF can be utilized to this end. The result will be reported as soon as it is available.

Third, the approach in this paper can be, at least formally, extended to the numerical integration of conservative/dissipative *partial* differential equations, for designing temporally high-order conservative/dissipative schemes. To do so, we first discretize space variable so that the resulting *ordinary* differential equations would be still conservative/dissipative. Then we apply the approach in this paper to the ordinary differential equations. But in this way we have to solve considerably high dimensional systems of nonlinear equations. We are now working on this issue, and will report some results in the near future.

Acknowledgement.

The authors are grateful to Prof. Masaaki Sugihara and Prof. Daisuke Furihata for their interest in this work and useful comments. They also would like to acknowledge Prof. Felice Iavernaro and Prof. Pierluigi Amodio for their valuable advices concerning the block BVM.

References

[1] Brugunano, L., and Trigiange, D., Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach Science Publishers, Amsterdam, 1998.

[2] Chorin, A.J., Hughes, McCracken, M.F. and Marsden, J.E., Product Formulas and Numerical Algorithms, *Comm. Pure and Appl. Math.* **31** (1978), 205–256.

[3] Furihata, D., Finite Difference Schemes for $\frac{\partial u}{\partial t} = \left(\frac{\partial}{\partial x}\right)^{\alpha} \frac{\delta G}{\delta u}$ that Inherit Energy Conservation or Dissipation Property, *J. Comput. Phys.* **156**(1999), 181–205.

[4] Furihata, D. and Matsuo, T., A Stable, Conservative, and Linear Finite Difference Scheme for the Cahn-Hilliard Equation, *Japan J. Indust. Appl. Math.*, **20** (2003), 65–85.

[5] Gonzalez, O., Time Integration and Discrete Hamiltonian Systems, *Nonlinear Science* **6** (1996), 449–467.

[6] Greenspan, D., Discrete Models, Addison-Wesley, Massachusetts, 1973.

[7] Hairer, E. and Lubich, C., Symmetric multistep methods over long times, *Numer. Math.*, **97** (2004), 699–723.

[8] Hairer, E., Lubich, C., and Wanner, G., Geometric Numerical Integration, Springer Verlag, 2002.

[9] Hairer, E., Nørsett, S.P. and Wanner, G., Solving Ordinary Differential Equations, I, 2nd edition, Springer Verlag, 1993.

[10] Iavernaro, F. and Mazzia, F., Generalization of backward differentiation formulas for parallel computers, *Numer. Algo.*, **31** (2002), 139–155.

[11] Matsuo, T. and Furihata, D., Dissipative or Conservative Finite Difference Schemes for Complex-Valued Nonlinear Partial Differential Equations, *J. Comput. Phys.* **171** (2001), 425–447.

[12] Matsuo, T., Sugihara, M., Furihata, D. and Mori, M., Spatially Accurate Dissipative or Conservative Finite Difference Schemes Derived by the Discrete Variational Method, *Japan J. Indust. Appl. Math.*, **19** (2003), 311–330.

[13] Matsuo, T., High-order schemes for conservative or dissipative systems, *J. Comput. Appl. Phys.* **152** (2003), 305–317.

[14] McLachlan, R.I., Quispel, G.R.W. and Robidoux, N., Geometric Integration using Discrete Gradients, *Phil. Trans. R. Soc. Lond. A* **357** (1999), 1021–1045.

[15] Yoshida, H., Construction of Higher Order Symplectic Integrators, *Phys. Lett. A* **150** (1990), 262–268.