



# On the optimization of $n$ -sub-step composite time integration methods

Huimin Zhang · Runsen Zhang ·  
Yufeng Xing · Pierangelo Masarati

Received: 12 May 2020 / Accepted: 12 October 2020  
© The Author(s) 2020

**Abstract** A family of  $n$ -sub-step composite time integration methods, which employs the trapezoidal rule in the first  $n - 1$  sub-steps and a general formula in the last one, is discussed in this paper. A universal approach to optimize the parameters is provided for any cases of  $n \geq 2$ , and two optimal sub-families of the method are given for different purposes. From linear analysis, the first sub-family can achieve  $n$ th-order accuracy and unconditional stability with controllable algorithmic dissipation, so it is recommended for high-accuracy purposes. The second sub-family has second-order accuracy, unconditional stability with controllable algorithmic dissipation, and it is designed for heuristic energy-conserving purposes, by preserving as much low-frequency content as possible. Finally, some illustrative examples are solved to check the performance in linear and nonlinear systems.

**Keywords**  $n$ -Sub-step composite method · Optimization · High-accuracy · Energy-conserving

## 1 Introduction

Direct time integration methods are frequently used to predict accurate numerical responses for general dynamic problems after spatial discretization. Driven by the pursuit of desirable properties, including higher accuracy and efficiency, robust stability, and many others, a number of excellent methods were proposed in the past decades.

In terms of the formulations, existing methods are generally classified into explicit and implicit schemes. Explicit methods are mostly used in wave propagation problems, as their conditional stability limits the allowable time step size to the highest system frequency. Implicit methods have fewer restrictions on the problems to be solved due to the unconditional stability, but they require more computational efforts per step.

In another way, the integration methods can also be divided into single-step, multi-sub-step and multi-step techniques. The single-step methods only adopt the states of the last step to predict the current one, while the multi-sub-step methods also need the states at the intermediate collocation points, and the multi-step methods require the states of more than one previous step. Each of them has specific advantages and disadvantages.

From the literature, representative single-step methods include the Newmark method [25], the HHT- $\alpha$  method (by Hilbert, Hughes, and Taylor) [17], the WBZ- $\alpha$  method (by Wood, Bossak, and Zienkiewicz) [29], the generalized- $\alpha$  method [9], the GSSS (gener-

---

Huimin Zhang · Runsen Zhang · Yufeng Xing  
School of Aeronautic Science and Engineering, Beihang  
University, Beijing 100083, China

Huimin Zhang (✉) · Runsen Zhang · Pierangelo Masarati  
Dipartimento di Scienze e Tecnologie Aerospaziali,  
Politecnico di Milano, 20156 Milan, Italy  
e-mail: huimin.zhang@polimi.it

alized single-step single-solve) method [34], and many others [28]. These single-step methods were proved to be spectrally identical to the linear multi-step methods [34], so they suffer from the Dahlquist's barrier [10], which states that the methods of higher than second-order accuracy cannot be unconditionally stable. Therefore, the methods mentioned above are all second-order accurate and unconditionally stable; some of them can also provide controllable algorithmic dissipation.

In the multi-step class, the Dahlquist's barrier certainly works, but in terms of accuracy, the linear two-step method [24, 33] is superior to most existing single-step methods under the same degree of algorithmic dissipation. In this class, BDFs (backward differentiation formulas) [11, 16] also represent a widely-used branch, particularly useful for stiff problems owing to the strong algorithmic dissipation. These popular multi-step methods are also second-order accurate and unconditionally stable. However, the multi-step methods are not self-starting, so another method has to be also used to solve the initial steps, which makes the multi-step methods not as convenient to use as the single-step ones.

The multi-sub-step methods, also known as multi-stage methods, allow more possibilities in terms of properties. The most representative method is the famous Runge–Kutta family [6, 7, 19], which can be designed to be arbitrarily higher-order accurate and unconditionally stable by choosing proper parameters and enough stages. Besides, Fung [12–15] provided some methods to reproduce the generalized Padé approximation. These methods can reach up to  $2n$ th-order accuracy by employing  $n$  sampling grid points per step, but the dimension of the implicit equation to be solved is  $n$  times that of the original, resulting in huge computational costs. In the multi-sub-step class, the composite methods [3], which divide each step into several sub-steps and employ different methods in each sub-step, have received a lot of attention in recent years.

Based on Bank et al.'s work [1], Bathe et al. [3] introduced the concept of the  $n$ -sub-step composite method by utilizing the trapezoidal rule in the first  $n - 1$  sub-steps and the  $(n + 1)$ -point backward difference scheme at the end of the step. The two-sub-step scheme is known as the Bathe method, which is asymptotically stable with second-order accuracy. Thanks to its strong dissipation and preferable accuracy, the Bathe method has been found to perform well in many fields [2, 4, 27]. The three-, and four-sub-

step composite methods [8, 32], which are asymptotically stable with higher accuracy, were also developed adopting the similar idea. Furthermore, to acquire controllable algorithmic dissipation, the two-sub-step methods [20, 21, 26], and the controllable three-sub-step methods [18, 23], were proposed by replacing the backward difference scheme with a more general formula. However, with the increase in the number of sub-steps, the number of scalar parameters required to be designed also increases, so the basic requirements, including second-order accuracy, unconditional stability, controllable algorithmic dissipation, are not enough to determine these parameters uniquely. Two optimal sub-families of the controllable three-sub-step method were proposed in [23], since different conditions are considered as a supplement.

On this basis, this paper purposes to provide a universal approach to optimize the parameters of generalized  $n$ -sub-step composite method, where  $n$  can be any integer greater than 2, and the trapezoidal rule is employed in the first  $n - 1$  sub-steps. Two kinds of optimization goals are considered, producing two optimal sub-families for different purposes. The first one intends to achieve higher-order accuracy, under the premises of unconditional stability and controllable algorithmic dissipation. The second one is dedicated to conserving low-frequency behavior, while still providing controllable high-frequency dissipation. From linear analysis, the resulting schemes in the first sub-family can reach up to  $n$ th-order accuracy by using  $n$  sub-steps, and the schemes in the second sub-family exhibit very small algorithmic dissipation in the low-frequency domain. Most of these schemes are developed for the first time, and in each sub-family, the accuracy can be improved by using more sub-steps. Finally, the proposed methods are applied to solve several numerical examples to check the performance.

This paper is organized as follows. The formulations of the  $n$ -sub-step composite method are shown in Sect. 2. The optimization of the parameters is implemented in Sect. 3. The detailed properties of the two sub-families are discussed in Sect. 4. Numerical examples are provided in Sect. 5, and conclusions are drawn in Sect. 6.

## 2 Formulation

In the literature, the composite methods were mostly developed to solve the problems in structural dynamics, as

$$M\ddot{\mathbf{x}} + \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t) = \mathbf{0}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad \dot{\mathbf{x}}(t_0) = \mathbf{v}_0 \quad (1)$$

where  $M$  is the mass matrix,  $F$  collects the damping force, internal force and external load,  $\mathbf{x}$ ,  $\dot{\mathbf{x}}$  and  $\ddot{\mathbf{x}}$  are the displacement, velocity and acceleration vectors, respectively,  $t$  is the time, and  $t_0$ ,  $\mathbf{x}_0$  and  $\mathbf{v}_0$  are the given initial time, displacement and velocity, respectively. When this method is applied using  $n$  sub-steps, it can be formulated as

$$M\ddot{\mathbf{x}}_{k+2j\gamma} + \mathbf{F}(\mathbf{x}_{k+2j\gamma}, \dot{\mathbf{x}}_{k+2j\gamma}, t_k + 2j\gamma h) = \mathbf{0} \quad (2a)$$

$$\mathbf{x}_{k+2j\gamma} = \mathbf{x}_{k+2(j-1)\gamma} + \gamma h (\dot{\mathbf{x}}_{k+2(j-1)\gamma} + \dot{\mathbf{x}}_{k+2j\gamma}) \quad (2b)$$

$$\dot{\mathbf{x}}_{k+2j\gamma} = \dot{\mathbf{x}}_{k+2(j-1)\gamma} + \gamma h (\ddot{\mathbf{x}}_{k+2(j-1)\gamma} + \ddot{\mathbf{x}}_{k+2j\gamma}) \quad (2c)$$

$$j = 1, 2, 3, \dots, n-1 \quad (2d)$$

and

$$M\ddot{\mathbf{x}}_{k+1} + \mathbf{F}(\mathbf{x}_{k+1}, \dot{\mathbf{x}}_{k+1}, t_k + h) = \mathbf{0} \quad (3a)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \left( \sum_{j=0}^{n-1} q_j \dot{\mathbf{x}}_{k+2j\gamma} + q_n \dot{\mathbf{x}}_{k+1} \right) \quad (3b)$$

$$\dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_k + h \left( \sum_{j=0}^{n-1} q_j \ddot{\mathbf{x}}_{k+2j\gamma} + q_n \ddot{\mathbf{x}}_{k+1} \right) \quad (3c)$$

where  $\mathbf{x}_k \approx \mathbf{x}(t_k)$  is the numerical solution at step  $k$ ,  $\mathbf{x}_{k+2j\gamma} \approx \mathbf{x}(t_k + 2j\gamma h)$  ( $j = 1, 2, \dots, n-1$ ) denotes the numerical solution at collocation points,  $h$  is the step size, and  $\gamma, q_0, q_1, \dots, q_n$  are the control parameters. The current step  $[t_k, t_k + h]$  is divided into  $n$  sub-steps:  $[t_k, t_k + 2\gamma h]$ ,  $[t_k + 2\gamma h, t_k + 4\gamma h]$ ,  $\dots$ ,  $[t_k + 2(n-2)\gamma h, t_k + 2(n-1)\gamma h]$ , and  $[t_k + 2(n-1)\gamma h, t_k + h]$ . In the first  $n-1$  sub-steps, the trapezoidal rule is adopted. In the last one, a general formula containing information about all collocation points is utilized. The present formulation can reduce to the  $\rho_\infty$ -Bathe method [26] when  $n = 2$  and to the three-sub-step method [18, 23] when  $n = 3$ .

In this method, because the same form of assumptions is used to solve  $\mathbf{x}_{k+1}$  and  $\dot{\mathbf{x}}_{k+1}$ , Eqs. (2) and (3)

can be reformulated based on the general first-order differential equation  $\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}, t) = \mathbf{0}$ , as

$$\mathbf{f}(\mathbf{y}_{k+2j\gamma}, \dot{\mathbf{y}}_{k+2j\gamma}, t_k + 2j\gamma h) = \mathbf{0} \quad (4a)$$

$$\mathbf{y}_{k+2j\gamma} = \mathbf{y}_{k+2(j-1)\gamma} + \gamma h (\dot{\mathbf{y}}_{k+2(j-1)\gamma} + \dot{\mathbf{y}}_{k+2j\gamma}) \quad (4b)$$

$$j = 1, 2, 3, \dots, n-1 \quad (4c)$$

and

$$\mathbf{f}(\mathbf{y}_{k+1}, \dot{\mathbf{y}}_{k+1}, t_k + h) = \mathbf{0} \quad (5a)$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h \left( \sum_{j=0}^{n-1} q_j \dot{\mathbf{y}}_{k+2j\gamma} + q_n \dot{\mathbf{y}}_{k+1} \right) \quad (5b)$$

where  $\{\mathbf{x}; \dot{\mathbf{x}}\}$  is replaced by  $\mathbf{y}$ , and the dynamics equations can be equivalently formulated as first-order differential equations by adding the trivial equation  $\dot{\mathbf{x}} = \dot{\mathbf{x}}$ . Equations (4) and (5) present more general formulations for solving first-order and arbitrarily higher-order differential equations. However, for solving the second-order dynamic problems, Eqs. (2) and (3) are still more recommended, since in the equivalent first-order expressions, the number of implicit equations to be solved doubles.

From the formulation, the first  $n-1$  sub-steps can share the same procedure in a loop, whereas the last sub-step needs to be implemented separately. The assumption  $q_n = \gamma$  is introduced here, which imposes that the last sub-step shares the same form of Jacobi matrix as the first  $n-1$  sub-steps. This assumption is particularly useful when applied to linear problems, since it allows the constant Jacobi matrix to be factorized only once, like in the single-step methods. For applications, Table 1 shows the computational procedures of the  $n$ -sub-step composite method for the general first-order differential equation  $\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}, t) = \mathbf{0}$ , where the Newton-Raphson iteration is utilized to solve the nonlinear equation per sub-step.

Besides, by reorganizing the formulations, the composite method can be regarded as a special case of the diagonally-implicit Runge–Kutta methods (DIRKs) with the explicit first-stage. The corresponding Butcher's tableau [6] has the form as

**Table 1** Computational procedure of the  $n$ -sub-step composite method for solving  $f(y, \dot{y}, t) = 0$ ,  $y(t_0) = y_0$ **A. Initial calculations**

1. From the function  $f(y, \dot{y}, t)$  and its derivative functions with respect to  $y$  and  $\dot{y}$ , as  $f_y$  and  $f_{\dot{y}}$ , respectively;
2. Initialize  $t_0$ ,  $y_0$  and  $\dot{y}_0$ ;
3. Select the time step size  $h$ , the algorithmic parameters  $\gamma$ ,  $q_0, q_1, \dots, q_{n-1}$ , the tolerance error  $\epsilon$ , and the maximum number of iterations  $N$ ;
4. Calculate the constant:  $a = \frac{1}{\gamma h}$ .

**B. For each time step**

1. The first  $n - 1$  sub-steps

For  $j = 1, j < n, j++$ :

- a. Predict  $y_{k+2j\gamma}$  and  $\dot{y}_{k+2j\gamma}$ :

$$i = 0, y_{k+2j\gamma} = y_{k+2(j-1)\gamma} + 2\gamma h \dot{y}_{k+2(j-1)\gamma}, \dot{y}_{k+2j\gamma} = a(y_{k+2j\gamma} - y_{k+2(j-1)\gamma}) - \dot{y}_{k+2(j-1)\gamma};$$

- b. Prepare the matrices:

$$f_{k+2j\gamma} = f(y_{k+2j\gamma}, \dot{y}_{k+2j\gamma}, t_k + 2j\gamma h), f_{y,k+2j\gamma} = f_y(y_{k+2j\gamma}, \dot{y}_{k+2j\gamma}, t_k + 2j\gamma h), f_{\dot{y},k+2j\gamma} = f_{\dot{y}}(y_{k+2j\gamma}, \dot{y}_{k+2j\gamma}, t_k + 2j\gamma h);$$

- c. Update  $y_{k+2j\gamma}$  and  $\dot{y}_{k+2j\gamma}$ :

$$i = i + 1, \Delta y_{k+2j\gamma} = -(f_{y,k+2j\gamma} + a f_{\dot{y},k+2j\gamma})^{-1} f_{k+2j\gamma}, y_{k+2j\gamma} = y_{k+2j\gamma} + \Delta y_{k+2j\gamma}, \dot{y}_{k+2j\gamma} = \dot{y}_{k+2j\gamma} + a \Delta y_{k+2j\gamma};$$

- d. If  $i < N$  and  $|f_{k+2j\gamma}| > \epsilon$ , go to b; If  $i \equiv N$  and  $|f_{k+2j\gamma}| > \epsilon$ , abort.

End.

2. The last sub-step

- a. Predict  $y_{k+1}$  and  $\dot{y}_{k+1}$ :

$$i = 0, y_{k+1} = y_{k+2(n-1)\gamma} + (1 - 2(n-1)\gamma)h \dot{y}_{k+2(n-1)\gamma}, \dot{y}_{k+1} = a(y_{k+1} - y_k - h \sum_{j=0}^{n-1} q_j \dot{y}_{k+2j\gamma});$$

- b. Prepare the matrices:

$$f_{k+1} = f(y_{k+1}, \dot{y}_{k+1}, t_k + h), f_{y,k+1} = f_y(y_{k+1}, \dot{y}_{k+1}, t_k + h), f_{\dot{y},k+1} = f_{\dot{y}}(y_{k+1}, \dot{y}_{k+1}, t_k + h);$$

- c. Update  $y_{k+1}$  and  $\dot{y}_{k+1}$ :

$$i = i + 1, \Delta y_{k+1} = -(f_{y,k+1} + a f_{\dot{y},k+1})^{-1} f_{k+1}, y_{k+1} = y_{k+1} + \Delta y_{k+1}, \dot{y}_{k+1} = \dot{y}_{k+1} + a \Delta y_{k+1};$$

- d. If  $i < N$  and  $|f_{k+1}| > \epsilon$ , go to b; If  $i \equiv N$  and  $|f_{k+1}| > \epsilon$ , abort.

0	0	0	0	$\dots$	0	0
$2\gamma$	$\gamma$	$\gamma$	0	$\dots$	0	0
$4\gamma$	$\gamma$	$2\gamma$	$\gamma$	$\ddots$	0	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$	$\vdots$
$2(n-1)\gamma$	$\gamma$	$2\gamma$	$2\gamma$	$\dots$	$\gamma$	0
1	$q_0$	$q_1$	$q_2$	$\dots$	$q_{n-1}$	$\gamma$
	$q_0$	$q_1$	$q_2$	$\dots$	$q_{n-1}$	$\gamma$

order differential equation is discussed, as

$$\dot{y} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\xi\omega \end{bmatrix} y, \quad y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (7)$$

Decomposing the coefficient matrix in Eq. (7) yields the simplified first-order equation

$$\dot{y} = \lambda y, \quad \lambda = (-\xi \pm i\sqrt{1 - \xi^2})\omega \quad (8)$$

where  $i = \sqrt{-1}$ . When the composite method is applied, the recursive scheme becomes

$$y_{k+1} = A(\lambda h) y_k \quad (9)$$

### 3 Optimization

In linear spectral analysis, owing to the mode superposition principle, it is common and enough to consider the single degree-of-freedom equation

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2x = 0 \quad (6)$$

where  $\xi$  is the damping ratio, and  $\omega$  is the natural frequency. To simplify the analysis, the equivalent first-

where the amplification factor  $A$  is

$$A(z) = (1 - q_n z)^{-1} \left( 1 + z \sum_{j=0}^{n-1} q_j \left( \frac{1 + \gamma z}{1 - \gamma z} \right)^j \right),$$

$$z = \lambda h \quad (10)$$

Since  $q_n = \gamma$  is assumed in Sect. 2, Eq. (10) is updated as

$$A(z) = \frac{(1 - \gamma z)^{n-1} + z \sum_{j=0}^{n-1} (q_j (1 + \gamma z)^j (1 - \gamma z)^{n-j-1})}{(1 - \gamma z)^n}$$

$$= \frac{1 + a_1 z + a_2 z^2 + \cdots + a_n z^n}{(1 - \gamma z)^n} \quad (11)$$

where the coefficient of  $z^p$  ( $p = 1, 2, \dots, n$ ) is represented by  $a_p$  ( $p = 1, 2, \dots, n$ ), expressed as

$$a_p = \binom{p}{n-1} (-\gamma)^p$$

$$+ \gamma^{p-1} \sum_{j=0}^{n-1} \left( q_j \sum_{m=\max\{0, p+j-n\}}^{\min\{j, p-1\}} P(m, j, p, n) \right),$$

$$P(m, j, p, n) = (-1)^{p-m-1} \binom{m}{j} \binom{p-m-1}{n-j-1},$$

$$p = 1, 2, \dots, n-1 \quad (12)$$

and

$$a_n = \gamma^{n-1} \sum_{j=0}^{n-1} \left( (-1)^{n-j-1} q_j \right) \quad (13)$$

For example,  $n = 5$  follows

$$a_1 = -4\gamma + q_0 + q_1 + q_2 + q_3 + q_4 \quad (14a)$$

$$a_2 = 6\gamma^2 + \gamma (-4q_0 - 2q_1 + 2q_3 + 4q_4) \quad (14b)$$

$$a_3 = -4\gamma^3 + \gamma^2 (6q_0 - 2q_2 + 6q_4) \quad (14c)$$

$$a_4 = \gamma^4 + \gamma^3 (-4q_0 + 2q_1 - 2q_3 + 4q_4) \quad (14d)$$

$$a_5 = \gamma^4 (q_0 - q_1 + q_2 - q_3 + q_4) \quad (14e)$$

Consequently, the parameters under analysis change from  $q_j$  ( $j = 0, 1, \dots, n-1$ ) and  $\gamma$ , to  $a_p$  ( $p = 1, 2, \dots, n$ ) and  $\gamma$  in the following. When  $a_p$  and  $\gamma$  are given, the parameters  $q_j$  can be obtained uniquely by solving Eqs. (12) and (13). For applications, Table 2 shows

**Table 2** Formulas of  $q_j$  ( $j = 0, 1, \dots, n-1$ )

$n$	$q_j$ ( $j = 0, 1, \dots, n-1$ )
2	$q_0 = \frac{\gamma}{2} + \frac{a_1}{2} - \frac{a_2}{2\gamma}$ $q_1 = \frac{\gamma}{2} + \frac{a_1}{2} + \frac{a_2}{2\gamma}$
3	$q_0 = \frac{3\gamma}{4} + \frac{a_1}{4} - \frac{a_2}{4\gamma} + \frac{a_3}{4\gamma^2}$ $q_1 = \gamma + \frac{a_1}{2} - \frac{a_3}{2\gamma^2}$ $q_2 = \frac{\gamma}{4} + \frac{a_1}{4} + \frac{a_2}{4\gamma} + \frac{a_3}{4\gamma^2}$
4	$q_0 = \frac{7\gamma}{8} + \frac{a_1}{8} - \frac{a_2}{8\gamma} + \frac{a_3}{8\gamma^2} - \frac{a_4}{8\gamma^3}$ $q_1 = \frac{11\gamma}{8} + \frac{3a_1}{8} - \frac{a_2}{8\gamma} - \frac{a_3}{8\gamma^2} + \frac{3a_4}{8\gamma^3}$ $q_2 = \frac{5\gamma}{8} + \frac{3a_1}{8} + \frac{a_2}{8\gamma} - \frac{a_3}{8\gamma^2} - \frac{3a_4}{8\gamma^3}$ $q_3 = \frac{\gamma}{8} + \frac{a_1}{8} + \frac{a_2}{8\gamma} + \frac{a_3}{8\gamma^2} + \frac{a_4}{8\gamma^3}$
5	$q_0 = \frac{15\gamma}{16} + \frac{a_1}{16} - \frac{a_2}{16\gamma} + \frac{a_3}{16\gamma^2} - \frac{a_4}{16\gamma^3} + \frac{a_5}{16\gamma^4}$ $q_1 = \frac{13\gamma}{8} + \frac{a_1}{4} - \frac{a_2}{8\gamma} + \frac{a_4}{8\gamma^3} - \frac{a_5}{4\gamma^4}$ $q_2 = \gamma + \frac{3a_1}{8} - \frac{a_3}{8\gamma^2} + \frac{3a_5}{8\gamma^4}$ $q_3 = \frac{3\gamma}{8} + \frac{a_1}{4} + \frac{a_2}{8\gamma} - \frac{a_4}{8\gamma^3} - \frac{a_5}{4\gamma^4}$ $q_4 = \frac{\gamma}{16} + \frac{a_1}{16} + \frac{a_2}{16\gamma} + \frac{a_3}{16\gamma^2} + \frac{a_4}{16\gamma^3} + \frac{a_5}{16\gamma^4}$

the formulas of  $q_j$  expressed by  $a_p$  and  $\gamma$  for the cases  $n = 2, 3, 4, 5$ .

### 3.1 Higher-order schemes

A numerical method is naturally expected to be as accurate as possible, so the higher-order schemes are considered first. From the scheme of Eq. (9), the composite method uses the amplification factor  $A$ , rewritten as

$$A(z) = \frac{1 + a_1 z + a_2 z^2 + \cdots + a_n z^n}{(1 - \gamma z)^n} \quad (15)$$

to approximate the exact amplification factor  $\hat{A}$

$$\hat{A}(z) = e^z = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \cdots \quad (16)$$

Hence the local truncation error  $\sigma$  can be defined as

$$\sigma = y_{k+1} - y(t_{k+1}) = (A(z) - \hat{A}(z))y(t_k) \quad (17)$$

If  $\sigma = O(z^{s+1})$ , the method is said to be  $s$ th-order accurate, which requires that up to  $s$ th derivatives of  $A$  at  $z = 0$  are all equal to 1, that is

$$A(0) = A^{(1)}(0) = A^{(2)}(0) = \cdots = A^{(s)}(0) = 1 \quad (18)$$

To satisfy Eq. (18),  $a_p$  ( $p = 1, 2, \dots, n$ ) can be solved as

$$A^{(1)}(0) = 1 \Rightarrow a_1 = 1 - n\gamma \quad (19a)$$

$$A^{(2)}(0) = 1 \Rightarrow a_2 = \frac{1}{2} - n\gamma + \frac{n(n-1)}{2}\gamma^2 \quad (19b)$$

$$A^{(3)}(0) = 1 \Rightarrow \quad (19c)$$

$$a_3 = \frac{1}{6} - \frac{n}{2}\gamma + \frac{n(n-1)}{2}\gamma^2 - \frac{n(n-1)(n-2)}{6}\gamma^3 \quad (19d)$$

$$A^{(s)}(0) = 1 \Rightarrow a_s = \sum_{j=0}^s \left( \frac{(-1)^j}{(s-j)!} \binom{j}{n} \gamma^j \right) \quad (19e)$$

Therefore, if all  $a_p$  ( $p = 1, 2, \dots, n$ ) follow the relationships in Eq. (19), this method can achieve  $n$ th-order accuracy, and then  $\gamma$  becomes the only free parameter to control the stability.

A time integration method is said to be unconditionally stable if  $|A(z)| \leq 1$  for all  $\Re(z) \leq 0$  where  $z = \lambda h = (-\xi \pm i\sqrt{1-\xi^2})\omega h$ . According to Ref. [19], the bounds on  $\gamma$  can be given by considering the stability on the imaginary axis ( $\xi = 0$ ), which can result in the unconditional stability when the accuracy order  $s = n$  in the DIRKs. Therefore, let  $z = \pm i\tau$  where  $\tau = \omega h$  is a real number, and

$$N(z) = 1 + a_1 z + a_2 z^2 + \cdots + a_n z^n \quad (20a)$$

$$D(z) = (1 - \gamma z)^n \quad (20b)$$

which are the numerator and denominator of  $A(z)$  in Eq. (15), respectively,  $|A(z)| \leq 1$  is equivalent to

$$|A(z)|^2 = A(i\tau)A(-i\tau) = \frac{N(i\tau)N(-i\tau)}{D(i\tau)D(-i\tau)} \leq 1 \quad (21)$$

Then the condition for unconditional stability can be transformed into

$$S(\tau) = D(i\tau)D(-i\tau) - N(i\tau)N(-i\tau) = \sum_{j=0}^n (c_{2j} \tau^{2j}) \geq 0 \text{ for } \tau \geq 0 \quad (22)$$

where the function  $S(\tau)$  is introduced, and the coefficients  $c_{2j}$  ( $j = 0, 1, 2, \dots, n$ ) are expressed as

$$c_{2j} = \binom{j}{n} \gamma^{2j} + (-1)^{j+1} \sum_{m=\max\{0, 2j-n\}}^{\min\{n, 2j\}} ((-1)^m a_m a_{2j-m}) \quad (23)$$

in which  $a_0$  is set to 1. By Eq. (22), the bounds on  $\gamma$  of the cases  $n = 2, 3, 4, 5$  are provided in Table 3. It follows that, with  $s = n$ , the allowable range of  $\gamma$  narrows as  $n$  increases and, in some cases, the  $n$ -sub-step method can achieve  $(n+1)$ th-order accuracy with a fixed  $\gamma$ .

Besides, algorithmic dissipation is also a desirable property for a time integration method, to filter out the inaccurate high-frequency content. Generally, it is measured by the spectral radius  $\rho_\infty$  at high-frequency limit, that is

$$|A(z)| \rightarrow \rho_\infty \text{ as } |z| = \omega h \rightarrow +\infty, \rho_\infty \in [0, 1] \quad (24)$$

and it gets stronger with a smaller  $\rho_\infty$ . With  $A(z)$  from Eq. (15), Eq. (24) can be satisfied if

$$a_n^2 = \left( \sum_{j=0}^n \left( \frac{(-1)^j}{(n-j)!} \binom{j}{n} \gamma^j \right) \right)^2 = \rho_\infty^2 \gamma^{2n} \quad (25)$$

which can be used to solve  $\gamma$  for a given  $\rho_\infty$ . Table 4 shows the solutions of  $\gamma$  for several specific  $\rho_\infty$  in

**Table 3** Bounds on  $\gamma$  for unconditional stability ( $s$  is the accuracy order) in the higher-order schemes

$n$	$s$	Bounds on $\gamma$
2	2	$[0.250000000000000, +\infty]$
	3	0.788675134594813
3	3	$[0.333333333333333, 1.068579021301628]$
	4	1.068579021301628
4	4	$[0.394337567297396, 1.280579761275305]$
	5	–
5	5	$[0.246505193142435, 0.361803398875471] \cup [0.420782512765729, 0.473268391258294]$
	6	0.473268391258294

the cases  $n = 2, 3, 4, 5$ . Note that Eq. (25) has multiple solutions; the smallest one that meets the requirement of unconditional stability, as shown in Table 3, is selected.

So far, the unconditionally stable higher-order accurate schemes with controllable algorithmic dissipation have been developed, whose parameter  $\gamma$  can be solved for a given  $\rho_\infty$  by Eq. (25),  $a_p$  ( $p = 1, 2, \dots, n$ ) are determined by  $\gamma$  as shown in Eq. (19), and then  $q_j$  ( $j = 0, 1, 2, \dots, n - 1$ ) can be obtained by solving Eqs. (12) and (13). These information for the cases  $n = 2, 3, 4, 5$  are shown in Tables 2, 3 and 4. The special case  $n = 2$  is identical to the  $\rho_\infty$ -Bathe method [26], whereas the other cases are presented here for the first time. In addition, the accuracy and algorithmic dissipation are discussed in more detail in Sect. 4.

### 3.2 Conserving schemes

An original intention of the composite methods was to conserve the energy of the system [2], which explains why the trapezoidal rule is utilized in most sub-steps. Existing two- and three-sub-step methods [18, 22, 26] really show preferable energy-conserving characteristic over other single- and multi-step methods. In this work, a simple and general approach to determine the parameters, which enable the  $n$ -sub-step composite method to conserve as much low-frequency content as possible, is proposed.

First of all, to be competitive, the method needs to have some basically useful properties, including at least second-order accuracy, which requires

$$a_1 = 1 - n\gamma \quad (26a)$$

$$a_2 = \frac{1}{2} - n\gamma + \frac{n(n-1)}{2}\gamma^2 \quad (26b)$$

and controllable algorithmic dissipation, achieved by

$$a_n^2 = \rho_\infty^2 \gamma^{2n} \quad (27)$$

In addition, unconditional stability also needs to be satisfied, which will be checked last.

To conserve the energy as much as possible, the spectral radius  $\rho = |A(z)|$  should be as close to 1 as possible over the low-frequency range. For the special case  $\rho_\infty = 1$ ,  $\rho$  should remain 1 in the whole frequency domain. For other cases  $0 \leq \rho_\infty < 1$ , the departure of  $\rho$  from unit value should be as slow as possible from  $\rho(0) = 1$ . Considering the conservative system ( $\xi = 0$ ), this purpose can be realized by making the function  $S(\tau)$ , defined in Eq. (22), as smooth as possible. It follows that  $S(0) = S^{(1)}(0) = S^{(2)}(0) = \dots = S^{(m)}(0) = 0$ , where  $S^{(m)}(0)$  is the  $m$ th-order derivative of  $S(\tau)$  at  $\tau = 0$ , and  $m$  should be as large as possible. As  $S(\tau)$  is a linear polynomial, the condition transforms into its coefficients  $c_{2j}$  ( $j = 0, 1, 2, \dots, m$ ) = 0. To clarify,  $c_{2j}$  ( $j = 0, 1, 2, \dots, n$ ) are enumerated as

$$c_0 = 0 \quad (28a)$$

$$c_2 = n\gamma^2 - a_1^2 + 2a_0a_2 \quad (28b)$$

$$c_4 = \frac{n(n-1)}{2}\gamma^4 - a_2^2 + 2a_1a_3 - 2a_0a_4 \quad (28c)$$

$$c_6 = \frac{n(n-1)(n-2)}{6}\gamma^6 - a_3^2 + 2a_2a_4 - 2a_1a_5 + 2a_0a_6 \quad (28d)$$

$$\dots \quad (28e)$$

$$c_{2n-2} = n\gamma^{2n-2} - a_{n-1}^2 + 2a_{n-2}a_n \quad (28f)$$



**Table 4**  $\gamma$  for controllable algorithmic dissipation in the higher-order schemes

$\rho_\infty$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
0.0	0.292893218813452	0.435866521508460	0.572816062482135	0.278053841136450
0.1	0.287089056989371	0.421486815409409	0.548366644975830	0.274141306031868
0.2	0.281754163448146	0.408500789512922	0.526386456842386	0.270459886774582
0.3	0.276820321671636	0.396647209121134	0.506330118970782	0.266978043925651
0.4	0.272233289109874	0.385731000460835	0.487797474812348	0.263670231711606
0.5	0.267949192431123	0.375602225015285	0.470480577621677	0.260515416607055
0.6	0.263932022500210	0.366142810103347	0.454130785036529	0.257496029856675
0.7	0.260151847569038	0.357257811967234	0.438536189902193	0.254597208170133
0.8	0.256583509747431	0.348869453074869	0.423503766067179	0.251806231183850
0.9	0.253205655191036	0.340912922771929	0.408841866120699	0.249112096529630
1.0	0.250000000000000	0.333333333333333	0.394337567297407	0.246505193142820

$$c_{2n} = \gamma^{2n} - a_n^2 \quad (28g)$$

From Eqs. (26) and (27), we can obtain  $c_2 = 0$  and  $c_{2n} = (1 - \rho_\infty^2)\gamma^{2n} \geq 0$ , respectively. For the case  $n = 2$ , the conditions on accuracy and algorithmic dissipation are enough to determine all parameters, resulting again in the  $\rho_\infty$ -Bathe method [26]. For other cases with  $n > 2$ , the  $n - 2$  remaining parameters,  $a_3, a_4, \dots, a_{n-1}$  and  $\gamma$ , are obtained by solving the equations  $c_4 = c_6 = \dots = c_{2n-2} = 0$ . The values of these parameters for the cases  $n = 3, 4, 5$  are shown in Table 5, where the set with  $\gamma$  close to  $\frac{1}{2n}$  is selected, which requires  $a_n = \rho_\infty \gamma^n$ .

Then all parameters of the conserving schemes have been given by combining Eqs. (12), (13), (26), (27) and  $c_4 = c_6 = \dots = c_{2n-2} = 0$ . The resulting scheme of  $n = 3$  is equivalent to the first sub-family of the three-sub-step method proposed in [23]; the other cases are presented here for the first time.

In particular, when  $\rho_\infty = 1$ , the resulting scheme is a  $n$ -sub-step method with the trapezoidal rule in all sub-steps, which is supposed to be unconditionally stable in the linear analysis. Empirically, the algorithmic dissipation is acquired by reducing the spectral radius  $\rho$ , so the dissipative schemes are likely to be also unconditionally stable, and even present more robust stability. For the undamped case ( $\xi = 0$ ), the stability can be guaranteed since  $S(\tau) = (1 - \rho_\infty^2)\gamma^{2n}\tau^{2n} \geq 0$ ; for other cases, the stability conditions of the schemes listed in Table 5 are checked one by one by considering  $\xi \in (0, 1]$  and  $\tau \in [0, 10000]$  numerically. As expected,  $\rho \leq 1$  is satisfied at every point in

all schemes, so these methods can be said to possess unconditional stability for linear problems. Other properties are discussed in Sect. 4.

## 4 Properties

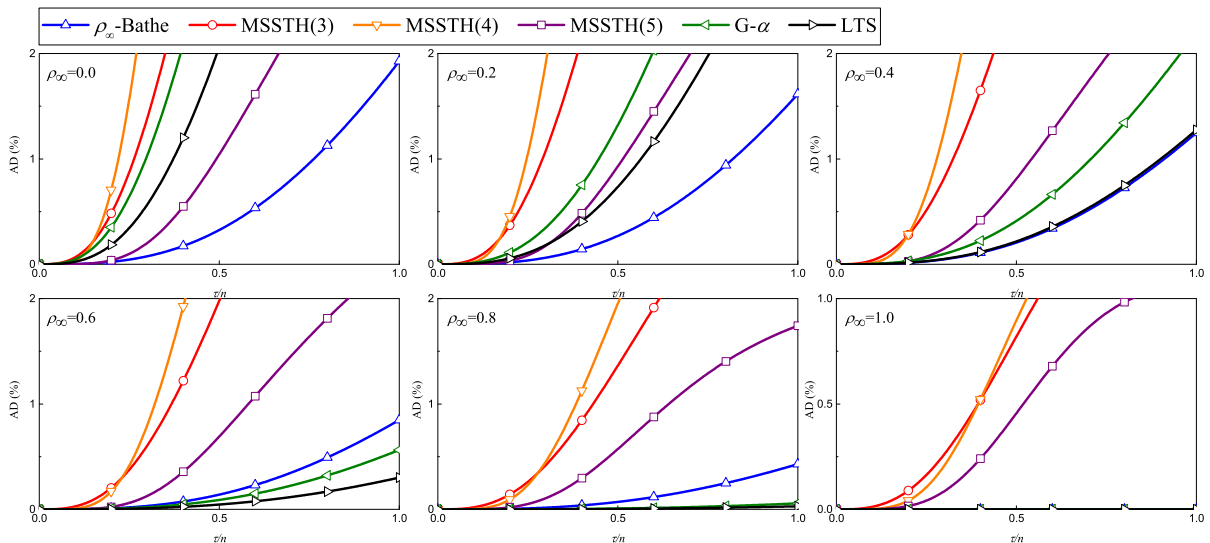
Two sub-families of the  $n$ -sub-step composite method have been presented for different purposes. To identify them, the higher-order schemes are referred to as MSSTH( $n$ ), and the conserving schemes are MSSTC( $n$ ), where MSST means the multi-sub-step composite method which employs the trapezoidal rule in all sub-steps except the last one, H and C are utilized to distinguish the two sub-families, and  $n$  is the number of sub-steps.

In this section, the representative methods in the literature, including the single-step generalized- $\alpha$  method [9] (G- $\alpha$ ) and the linear two-step method [33] (LTS) are also considered for comparison. As the employed methods are all implicit, their computational cost is mainly spent on the iterative calculation when used for non-linear problems, or the matrix factorization for linear problems. The vector operations brought by the recursive scheme of the method itself is generally considered to have little effect on overall efficiency. Therefore, G- $\alpha$  and LTS are recognized as having equivalent efficiency if the same step size is used. As the composite methods implement a single-step or multi-step scheme in each sub-step, they have the equivalent efficiency to G- $\alpha$  and LTS, if their required number of sub-steps is equal to the number of steps required by G- $\alpha$  and LTS. For this



**Table 5**  $a_p$  ( $p = 3, 4, \dots, n-1$ ) and  $\gamma$  for controllable algorithmic dissipation in the conserving schemes

$\rho_\infty$	$n = 3$ $\gamma$	$n = 4$ $a_3$	$\gamma$	$n = 5$ $a_3$	$a_4$	$\gamma$
0.0	0.180425306429398	0.00453529185986996	0.131378736730466	0.00763819606391975	0.000257160742971488	0.103557108920215
0.1	0.178619458204658	0.00494493283913114	0.130548620946472	0.00793598250555122	0.000286872844754286	0.103095631511675
0.2	0.176945806618224	0.00532986673141648	0.129777583818848	0.00821506063336095	0.000314958374958848	0.102666675025093
0.3	0.175385515842846	0.00569340893903897	0.129057207257355	0.00847799948547669	0.000341631205811411	0.102265594492185
0.4	0.173923607877197	0.00603821324069812	0.128380804919945	0.00872684827484184	0.000367062789628561	0.101888703879882
0.5	0.172547961422089	0.00636644119939074	0.127742970556848	0.00896327074168002	0.000391393000239752	0.101533025147874
0.6	0.171248618590691	0.00667987987582935	0.127139265902084	0.00918863822587218	0.000414737666914113	0.101196115073181
0.7	0.170017291772476	0.00698002584685393	0.126565999083137	0.00940409590940791	0.000437193952981663	0.100875942445807
0.8	0.168847004679168	0.00726814657645952	0.126020063586496	0.00961061107396470	0.000458844285976519	0.100570798918745
0.9	0.167731825756887	0.00754532615554921	0.125498818830422	0.00980900897660744	0.000479759288032603	0.100279232954742
1.0	0.166666666666667	0.00781250000000000	0.125000000000000	0.0100000000000000	0.00050000000000000	0.100000000000000

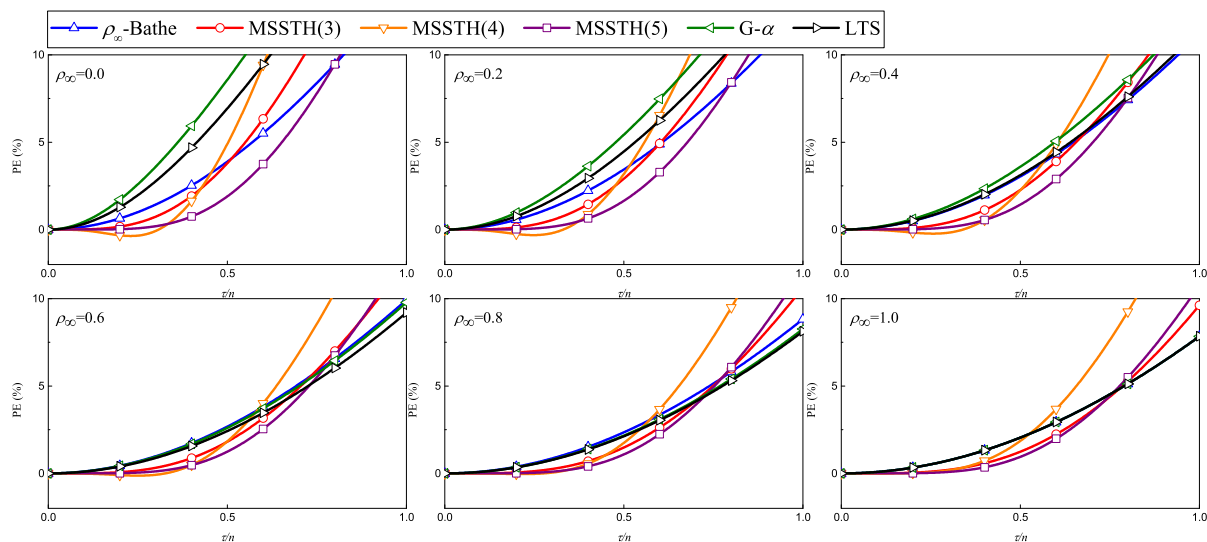

**Fig. 1** Percentage amplitude decay for MSSTH(2,3,4,5),  $G-\alpha$  and LTS

reason, to compare the properties under the close computational costs, the same  $h/n$ , where  $n$  is the number of sub-steps in the composite methods, and  $n = 1$  for the  $G-\alpha$  and LTS, is used in these methods.

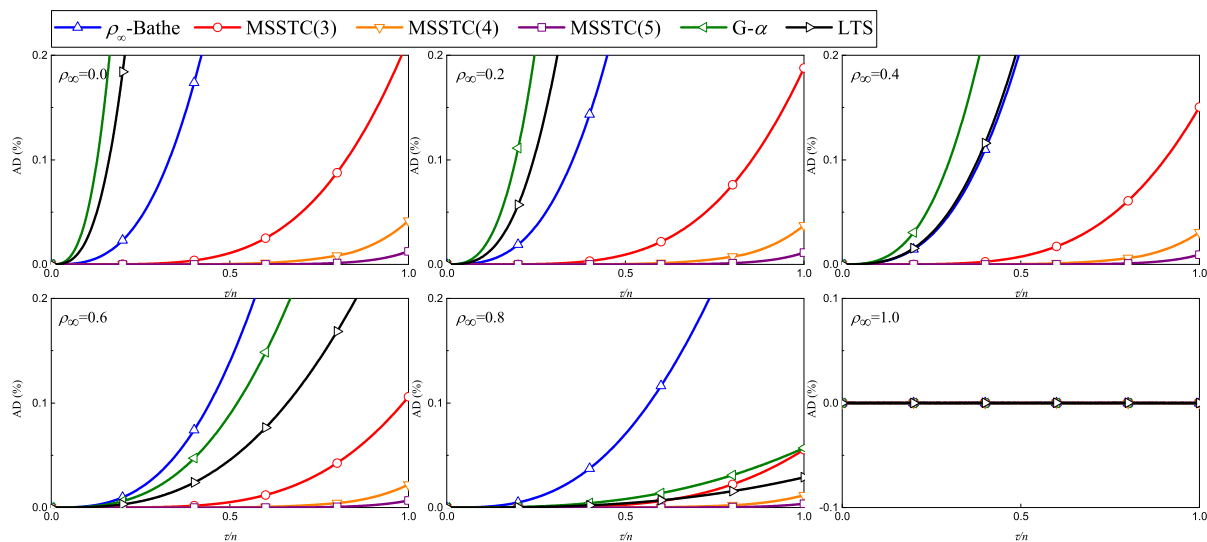
As discussed in Sect. 3, MSSTH( $n$ ) has  $n$ th-order accuracy under the premises of unconditional stability and controllable algorithmic dissipation. Figures 1 and 2 display the percentage amplitude decay (AD(%)) and period elongation (PE(%)) respectively, of which the definition can refer to [34], of MSSTH(2,3,4,5),  $G-\alpha$  and LTS, considering the undamped case ( $\xi = 0$ ). The abscissa is set as  $\tau/n$  to compare these methods under the close computational costs.

The results illustrate that the amplitude and period accuracy cannot be improved simultaneously as the order of accuracy increases in MSSTH( $n$ ). In terms of amplitude, with a small  $\rho_\infty$ , the  $\rho_\infty$ -Bathe method (the same as MSSTH(2)) is the most accurate, and when  $0.4 < \rho_\infty \leq 1$ , LTS shows smaller dissipation error, followed by the  $G-\alpha$  and the  $\rho_\infty$ -Bathe method. From Fig. 2, MSSTH(3,4,5) have smaller period error than the second-order methods, and MSSTH(5) is the best among them.

In the same way, the percentage amplitude decay and period elongation of MSSTC(2,3,4,5),  $G-\alpha$  and LTS for the undamped case are shown in Figs. 3 and 4,



**Fig. 2** Percentage period elongation for MSSTH(2,3,4,5), G- $\alpha$  and LTS



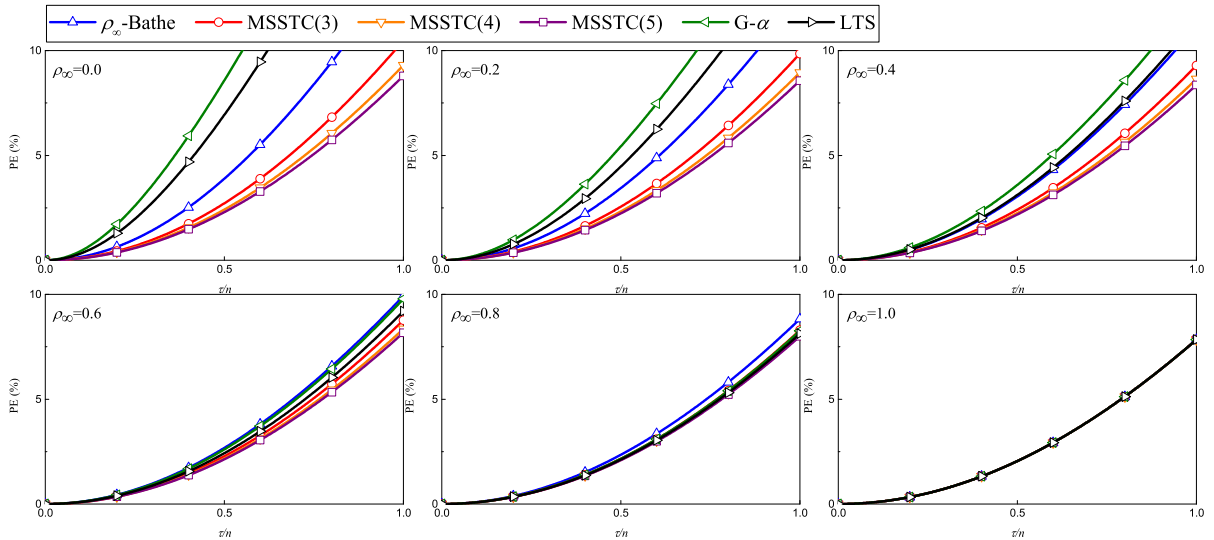
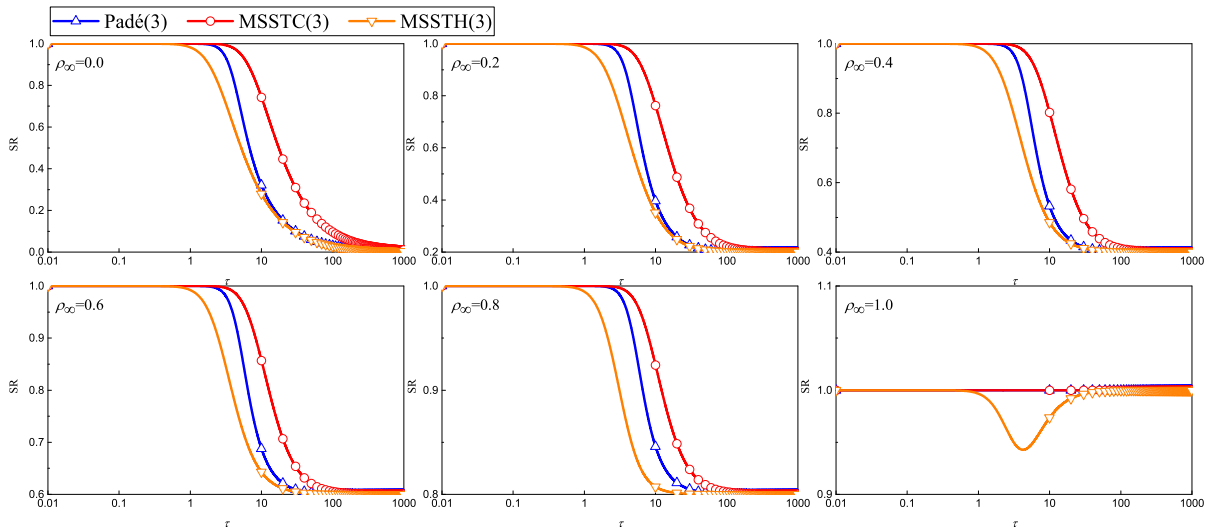
**Fig. 3** Percentage amplitude decay for MSSTC(2,3,4,5), G- $\alpha$  and LTS

respectively. It can be observed that under the similar efficiency, MSSTC( $n$ ) presents higher amplitude and period accuracy with a larger  $n$ . The gap is more obvious as  $\rho_\infty$  decreases, and when  $\rho_\infty = 1$ , all the schemes have the same properties as the trapezoidal rule. Both G- $\alpha$  and LTS are less accurate than MSSTC(3,4,5) in the low-frequency range.

Besides, with the same  $n$ , MSSTH( $n$ ) and MSSTC( $n$ ) are compared in Figs. 5, 6, 7, 8, 9, 10, 11, 12 and 13, where Figs. 5, 6 and 7 show the spectral radius (SR)

of the cases  $n = 3, 4, 5$ , respectively, Figs. 8, 9 and 10 show the percentage amplitude decay, Figs. 11, 12 and 13 show the percentage period elongation, all considering the undamped case. The generalized Padé approximation [14, 15], referred to as Padé( $n$ ), is also employed for comparison. It is known as the most accurate rational approximation of  $e^z$  by using

$$A(z) = \frac{(1 - \rho_\infty)P_{n-1,n}(z) + 2\rho_\infty P_{n,n}(z)}{(1 - \rho_\infty)Q_{n-1,n}(z) + 2\rho_\infty Q_{n,n}(z)} \quad (29)$$


**Fig. 4** Percentage period elongation for MSSTC(2,3,4,5),  $G-\alpha$  and LTS

**Fig. 5** Spectral radius for  $n = 3$ 

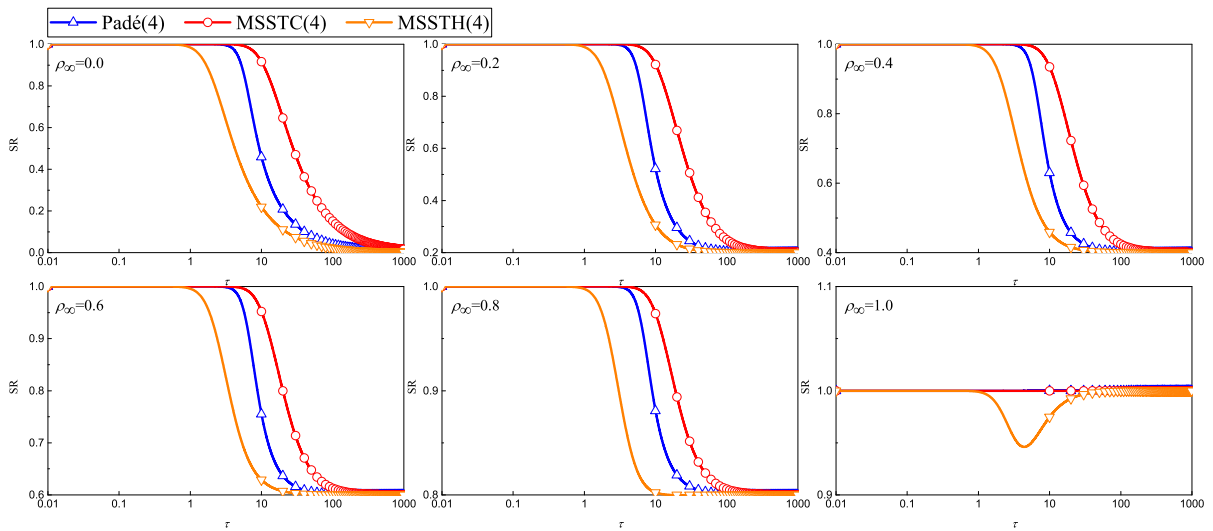
where

$$P_{i,j}(z) = \sum_{p=0}^i \frac{i!(j+i-p)!}{(i-p)!(j+i)!} \frac{z^p}{p!} \quad (30a)$$

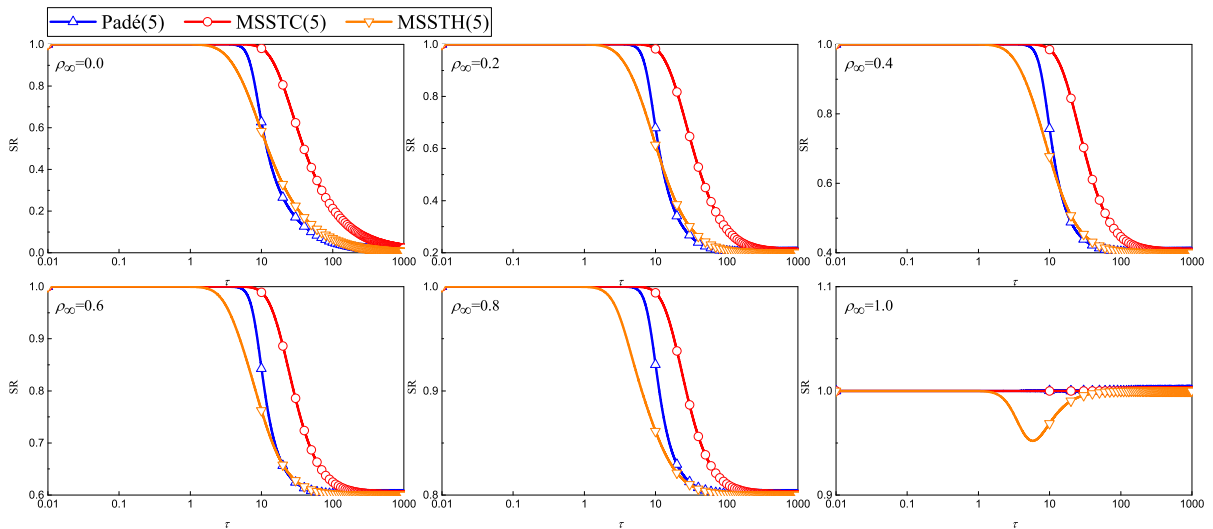
$$Q_{i,j}(z) = \sum_{p=0}^i (-1)^p \frac{j!(j+i-p)!}{(j-p)!(j+i)!} \frac{z^p}{p!} \quad (30b)$$

Padé( $n$ ) has  $(2n-1)$ th-order accuracy if  $0 \leq \rho_\infty < 1$  and  $(2n)$ th-order accuracy if  $\rho_\infty = 1$ .

As expected, Figs. 5, 6 and 7 demonstrate that MSSTC( $n$ ) preserves wider low-frequency range, followed by Padé( $n$ ), and MSSTH( $n$ ). Note that MSSTH( $n$ ) with  $\rho_\infty = 1$  exhibits mild algorithmic dissipation in the medium frequency range, so these schemes are not recommended if all frequencies are requested. Figures 8, 9 and 10 also show that MSSTC( $n$ ) has the smallest amplitude dissipation in the low-frequency content. The amplitude decay ratio of MSSTC(5) is very close to 0 over  $\tau \in [0, 2]$ . In terms of period accuracy, Figs. 11, 12 and 13 show that Padé( $n$ ) is the most



**Fig. 6** Spectral radius for  $n = 4$



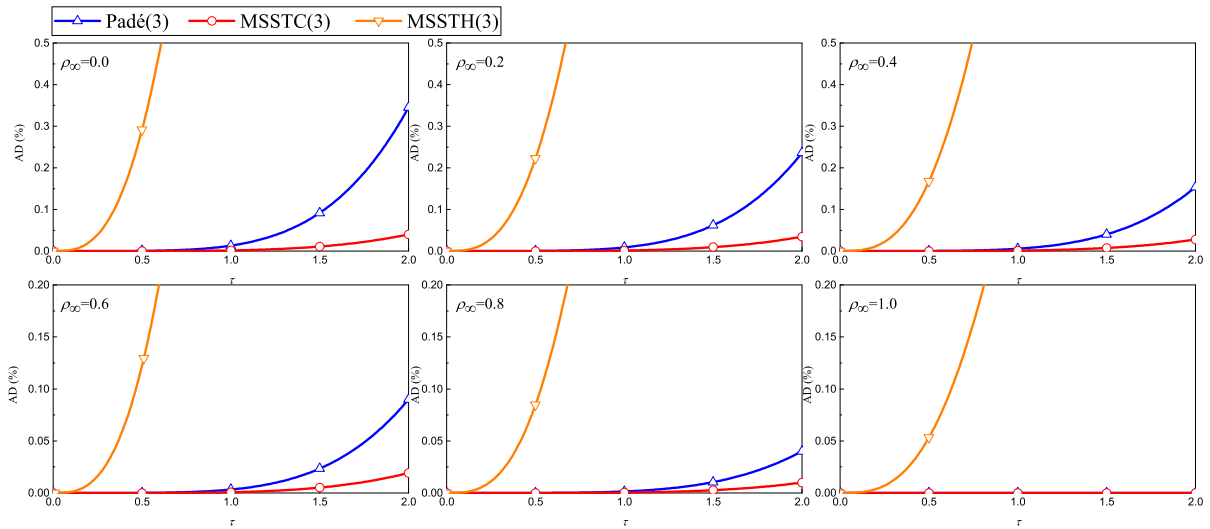
**Fig. 7** Spectral radius for  $n = 5$

accurate, followed by  $MSSTH(n)$ , and  $MSSTC(n)$ , consistent with the sequence of the accuracy order.

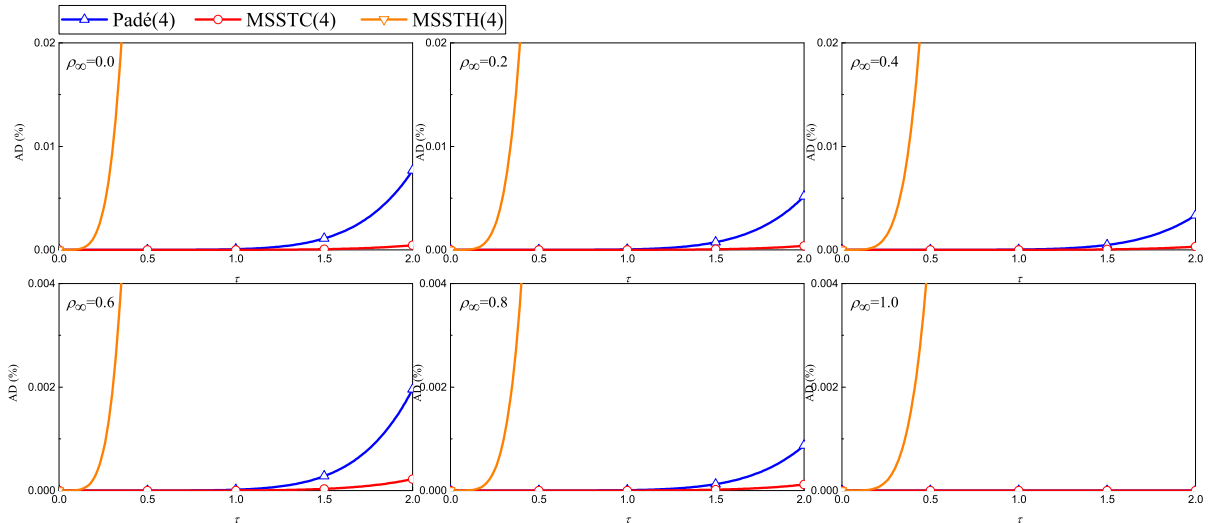
From the comparison,  $MSSTC(n)$  performs really good at conserving the low-frequency content, and its overall accuracy can be improved by using more substeps.  $MSSTH(n)$  shows higher period accuracy than the second-order methods, whereas its dissipation error is larger in the low-frequency content.

## 5 Numerical examples

To validate the performance, several numerical examples are solved in this section. As the spectral analysis has revealed the properties based on the linear model, this section focuses more on the application and discussion for nonlinear systems.



**Fig. 8** Percentage amplitude decay for  $n = 3$



**Fig. 9** Percentage amplitude decay for  $n = 4$

### 5.1 Single degree-of-freedom examples

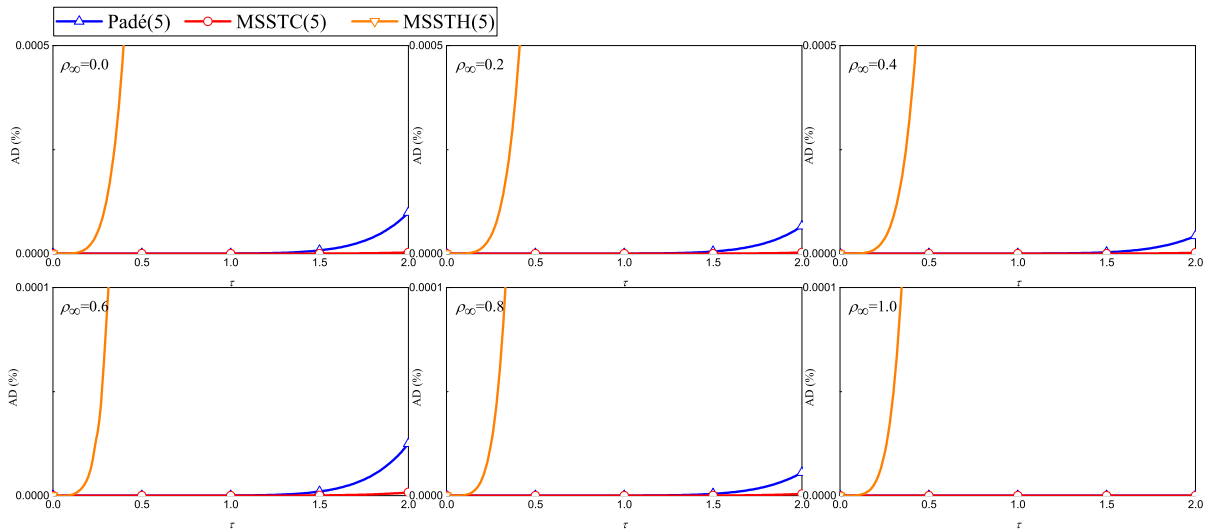
Firstly, two single degree-of-freedom examples, including a simple linear example and the nonlinear van der Pol's equation, are solved to check the convergence rate. The  $\rho_\infty$ -Bathe method, MSSTC(3,4,5) and MSSTH(3,4,5) with  $\rho_\infty = 0.6$  is employed.

*Linear example* The linear equation of motion

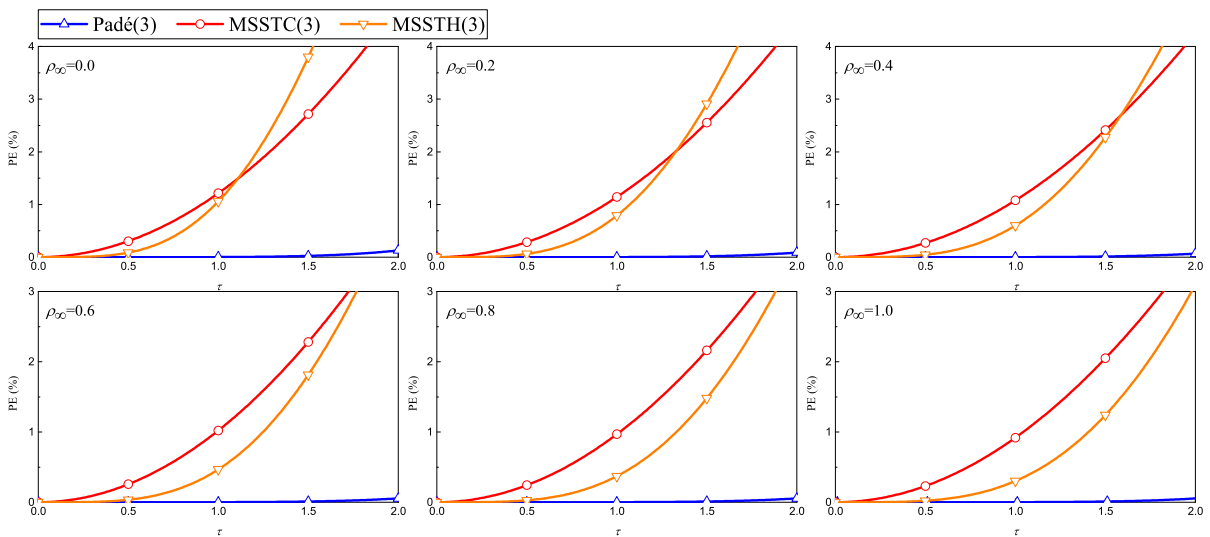
$$\ddot{x} + 4x = 0, x(0) = 1, \dot{x}(0) = 1 \quad (31)$$

is considered, and the absolute errors of the displacement  $x_k$ , velocity  $\dot{x}_k$ , and acceleration  $\ddot{x}_k$  versus  $h$  at  $t = 10$  are plotted in Fig. 14.

The results are consistent with the accuracy order. That is, MSSTC( $n$ ) and MSSTH( $n$ ) respectively present second-order and  $n$ th-order convergence rate. As a result, the higher-order MSSTH( $n$ ) enjoys significant accuracy advantage over the second-order methods. However, when  $h$  decreases from  $10^{-2}$ , it seems that MSSTH(5) cannot maintain fifth-order accuracy. This is because when  $h$  is small enough, all effective num-



**Fig. 10** Percentage amplitude decay for  $n = 5$



**Fig. 11** Percentage period elongation for  $n = 3$

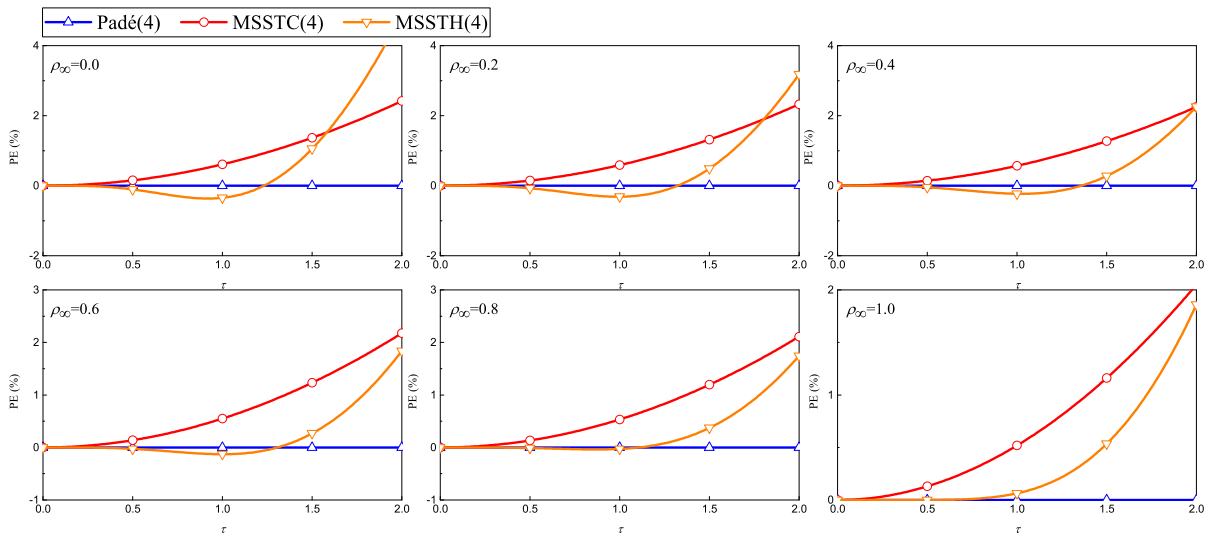
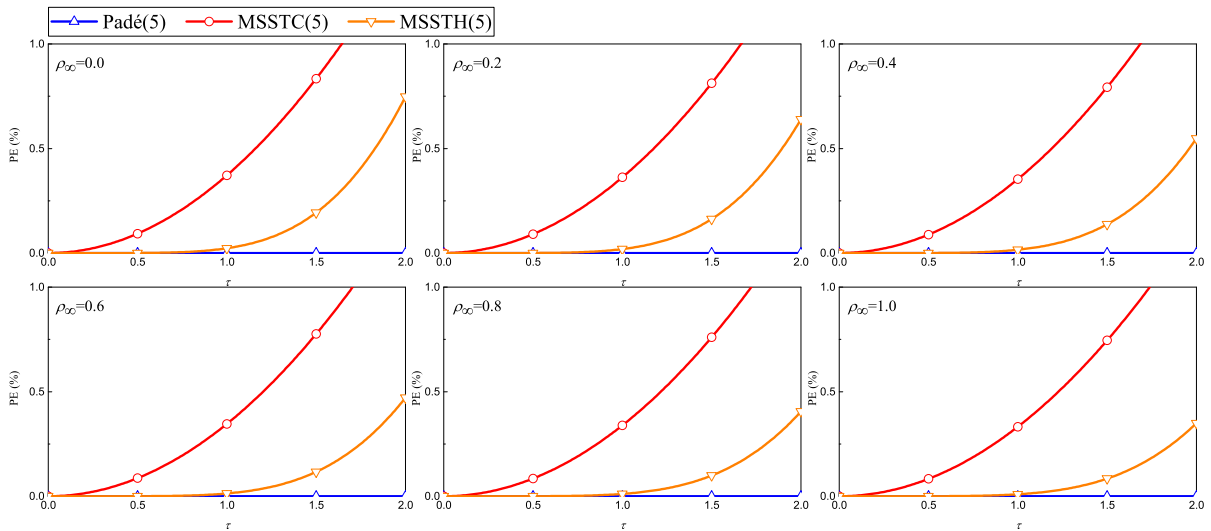
bers stored in the computer are exactly precise, so if  $h$  continues to decrease, the accumulated rounding error can greatly spoil the numerical precision [31].

**Van der Pol's equation** The van der Pol's equation [19]

$$\begin{aligned} \dot{x}_1 &= x_2, \dot{x}_2 = \epsilon^{-1}((1 - x_1^2)x_2 - x_1) \\ x_1(0) &= 2, x_2(0) = -\frac{2}{3} + \frac{10}{81}\epsilon - \frac{292}{2187}\epsilon^2 + \frac{15266}{59049}\epsilon^3 \end{aligned} \quad (32)$$

is solved, where  $\epsilon$  is an adjustable parameter. For the cases  $\epsilon = 0.01, 0.001, 0.0001$ , the absolute errors of  $x_{1,k}$  and  $x_{2,k}$  at  $t = 1$  versus  $h$  are plotted in Fig. 15, where the reference solution is obtained by the  $\rho_\infty$ -Bathe method with  $h = 10^{-7}$ .

From Fig. 15, in most cases, the second- and  $n$ th-order convergence rate can be observed from errors of MSSTC( $n$ ) and MSSTH( $n$ ), respectively, but for the stiffer case of  $\epsilon = 0.0001$ , MSSTH(3) and MSSTH(5) show obvious order reduction in both  $x_1$  and  $x_2$ . It indi-

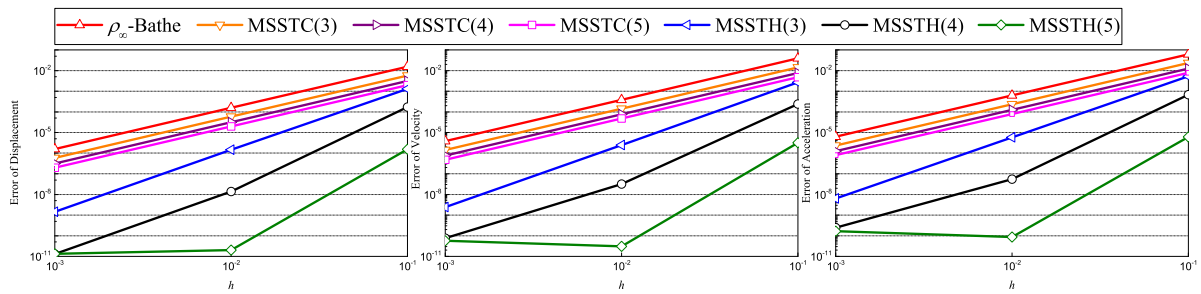

**Fig. 12** Percentage period elongation for  $n = 4$ 

**Fig. 13** Percentage period elongation for  $n = 5$ 

cates that the accuracy order also depends on the problem to be solved when applied to nonlinear systems. The order reduction also occurs in other higher-order DIRKs when used for nonlinear problems, see Ref. [19]. Nevertheless, MSSTH( $n$ ) still shows significant accuracy advantage over the second-order MSSTC( $n$ ) with a small step size.

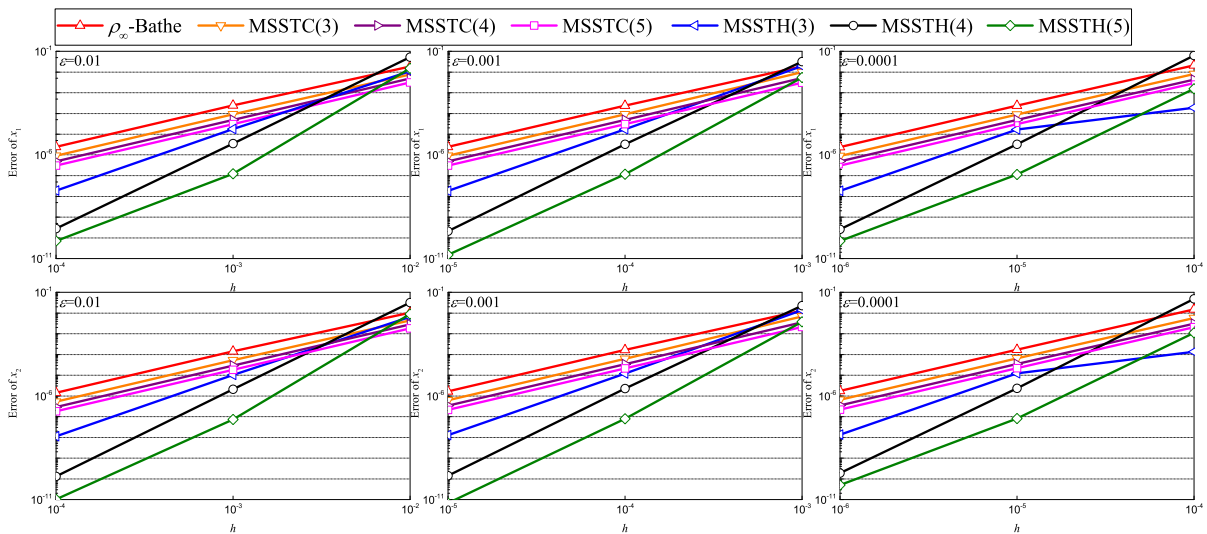
## 5.2 Multiple degrees-of-freedom examples

In this subsection, some illustrative examples are solved by using the  $\rho_\infty$ -Bathe method, MSSTC(3,4,5), MSSTH(3,4,5), G- $\alpha$  and LTS. In these methods, the parameter  $\rho_\infty$  is set as 0 uniformly, and the same  $h/n$  is used for comparison under close computational costs. The reference solutions are obtained by the  $\rho_\infty$ -Bathe method with an extremely small time step.

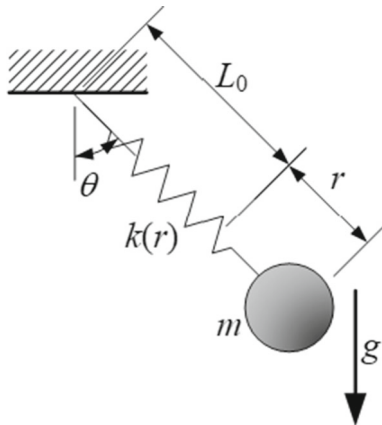




**Fig. 14** Convergence rates for the single degree-of-freedom linear example



**Fig. 15** Convergence rates for the van der Pol's equation



**Fig. 16** Spring-pendulum model

*Spring-pendulum model* As shown in Fig. 16, the spring-pendulum model, where the spring is fixed at

one end and with a mass at the free end, is simulated. Its motion equation can be written as

$$m\ddot{r} + f(r) - m(L_0 + r)\dot{\theta}^2 - mg \cos \theta = 0 \quad (33a)$$

$$m\ddot{\theta} + \frac{m(2\dot{r}\dot{\theta} + g \sin \theta)}{L_0 + r} = 0 \quad (33b)$$

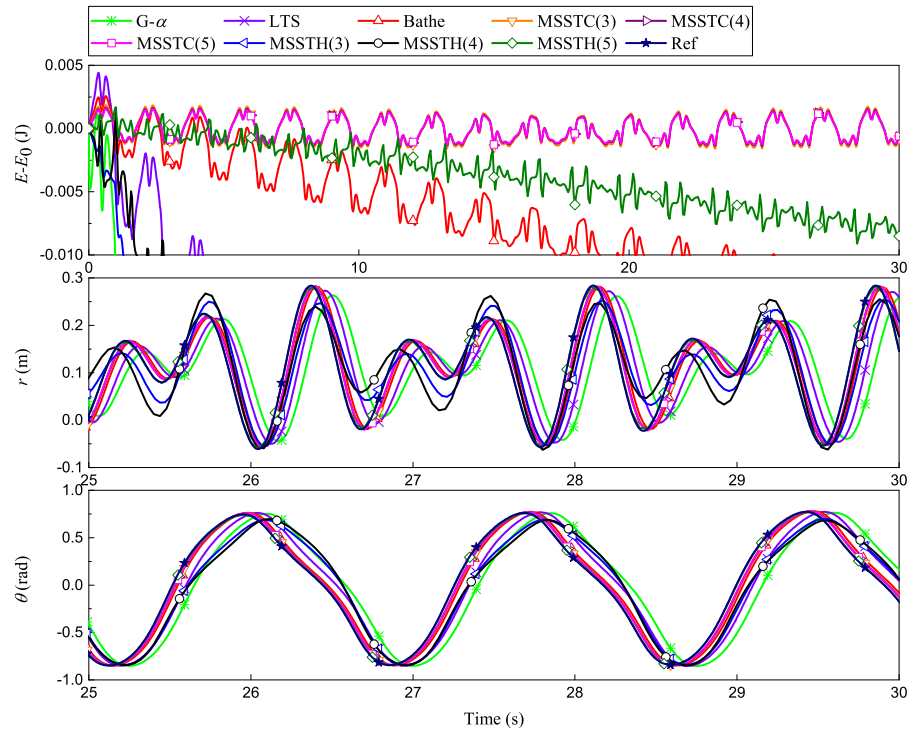
where  $f(r)$  denotes the elastic force of the spring and other system parameters are assumed as  $m = 1$  kg,  $L_0 = 0.5$  m,  $g = 9.81$  m/s<sup>2</sup>. Three kinds of constitutive relations, as

$$f(r) = kr \quad (34a)$$

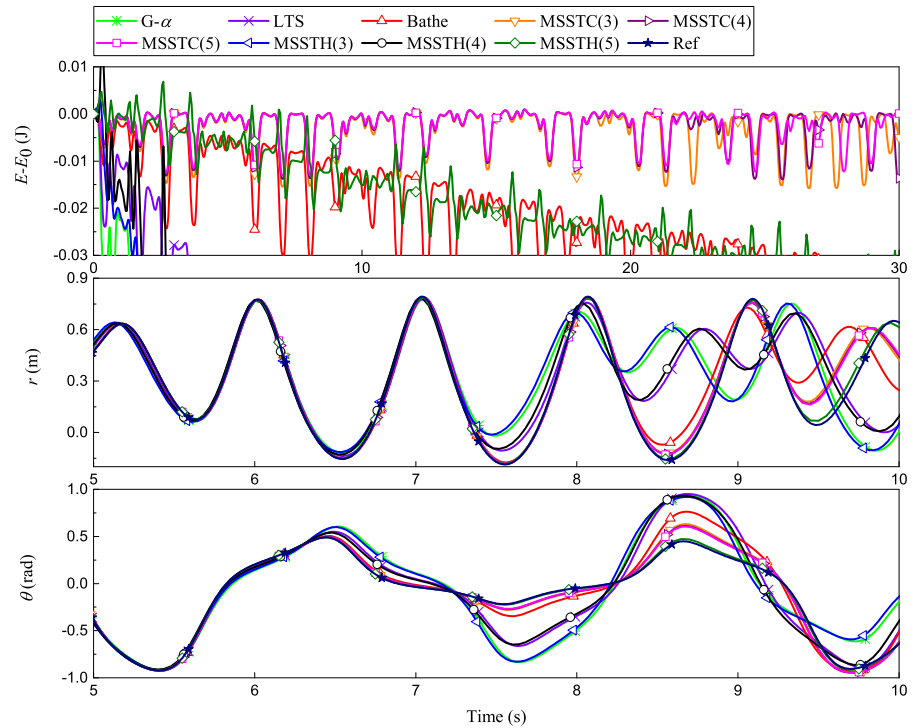
$$f(r) = kr^3 \quad (34b)$$

$$f(r) = k \tanh r \quad (34c)$$

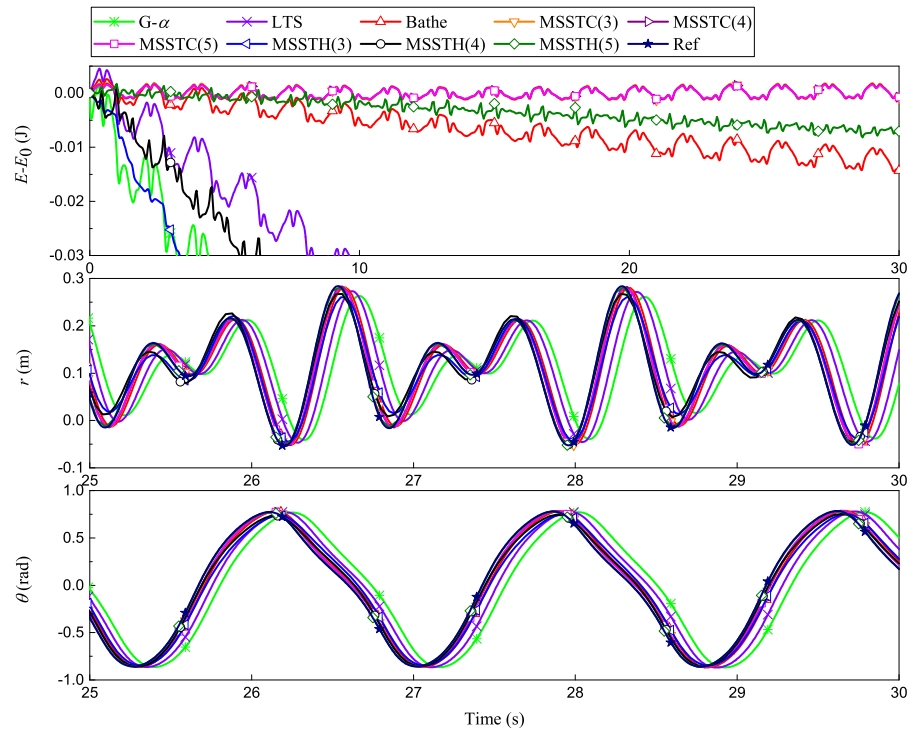
**Fig. 17** Numerical results of the spring-pendulum model ( $f(r) = kr$ ,  $k = 98.1$  N/m)



**Fig. 18** Numerical results of the spring-pendulum model ( $f(r) = kr^3$ ,  $k = 98.1$  N/m)



**Fig. 19** Numerical results of the spring-pendulum model ( $f(r) = k \tanh r$ ,  $k = 98.1 \text{ N/m}$ )



where  $k = 98.1 \text{ N/m}$ , are considered. The initial conditions are set as

$$r_0 = 0 \text{ m}, \dot{r}_0 = 1 \text{ m/s}, \theta_0 = \frac{\pi}{4} \text{ rad}, \dot{\theta}_0 \text{ rad/s} \quad (35)$$

Let  $h/n = 0.01 \text{ s}$ ; the numerical solutions of  $E - E_0$  ( $E$  denotes the system energy and  $E_0$  is the initial value),  $r$  and  $\theta$  for the three cases are summarized in Figs. 17, 18 and 19. From the curves of  $E - E_0$ , it can be observed that MSSTC(3,4,5) can almost preserve the numerical energy from decaying in all cases, despite the oscillations. MSSTH(5) can preserve more energy than the Bathe method, while  $G-\alpha$ , LTS, and MSSTH(3,4) show obvious energy-decaying. From the numerical results of  $r$  and  $\theta$ , one can see that with the step size, the numerical solutions of these methods have clearly deviated from the reference solution after a period of simulation. Among these methods, MSSTH(5) predicts the closest solutions to the reference ones, and  $G-\alpha$  shows the largest errors. In addition, MSSTC(3,4,5) exhibit good amplitude accuracy thanks to their energy-preserving characteristic. These conclusions are all consistent with the results from linear analysis.

Moreover, to check the algorithmic dissipation, the stiff case, where  $f(r) = kr$  ( $k = 98.1 \times 10^{10} \text{ N/m}$ ), is also simulated with  $h/n = 0.01 \text{ s}$ . The numerical results of  $E - E_0$ ,  $r$  and  $\theta$  are plotted in Fig. 20. The results of  $r$  indicate that all employed schemes with  $\rho_\infty = 0$  can effectively filter out the stiff component in the first few steps. After the initial decaying, MSSTC(3,4,5) can still preserve the remaining energy in the following simulation.

*Slider-pendulum model* The slider-pendulum model, shown in Fig. 21, is considered in this case. The slider is constrained by the spring, and one end of the pendulum is hinged to the center of mass of the slider. The motion is described by the differential-algebraic equations

$$m_1 \ddot{x}_1 + kx_1 = -\lambda_1 \quad (36a)$$

$$m_2 \ddot{x}_2 = \lambda_1 \quad (36b)$$

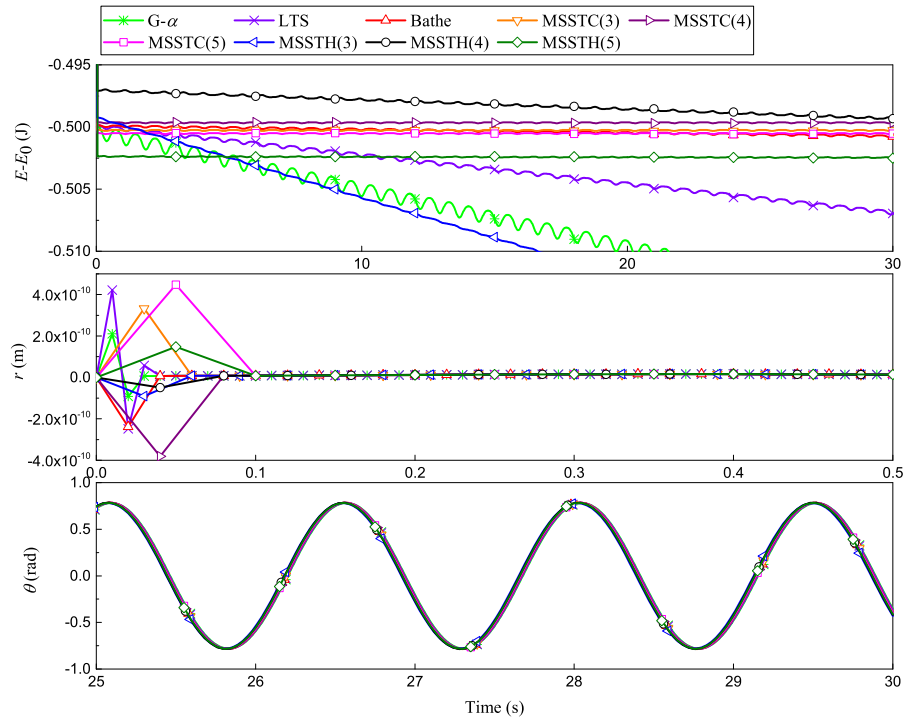
$$m_2 \ddot{y}_2 = \lambda_2 - m_2 g \quad (36c)$$

$$J_2 \ddot{\theta} = -\frac{L}{2} \lambda_1 \cos \theta - \frac{L}{2} \lambda_2 \sin \theta \quad (36d)$$

$$x_2 - x_1 = \frac{L}{2} \sin \theta \quad (36e)$$

$$y_2 = -\frac{L}{2} \cos \theta \quad (36f)$$

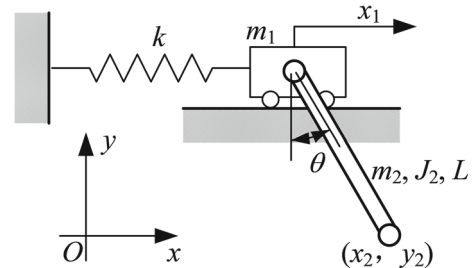
**Fig. 20** Numerical results of the spring-pendulum model ( $f(r) = kr$ ,  $k = 98.1 \times 10^{10}$  N/m)



The system parameters are  $m_1 = m_2 = 1$  kg,  $L = 1$  m,  $J_2 = \frac{1}{12}$  kg  $\cdot$  m $^2$ ,  $g = 9.81$  m/s $^2$ ,  $k = 1$  N/m and  $10^{10}$  N/m respectively for the compliant and stiff systems. The slider is excited by the initial horizontal velocity 1 m/s.

By using  $h/n = 0.01$  s, the numerical solutions of  $E - E_0$ ,  $x_1$  and  $\theta$  for the compliant and stiff cases are shown in Figs. 22 and 23, respectively. From the results of  $x_1$  and  $\theta$ , these methods all perform well in terms of accuracy and algorithmic dissipation. However, the numerical energies of MSSTH(4) show a slightly upward trend in the stiff case, so this method cannot give stable results for the problem.

As already discussed in several papers [5,30], the unconditional stability of a time integration method derived from linear analysis cannot be guaranteed when they are applied to nonlinear problems. For nonlinear problems, the stability of a method depends not only on its recursive scheme, but also on the problem itself. Therefore, it is hard to give a definite conclusion about the stability of a method for general problems. From the numerical results, all employed methods, except MSSTH(4), provide stable results when solving stiff problems and differential-algebraic equations, so they can be said to have relatively strong stability.



**Fig. 21** Slider-pendulum model

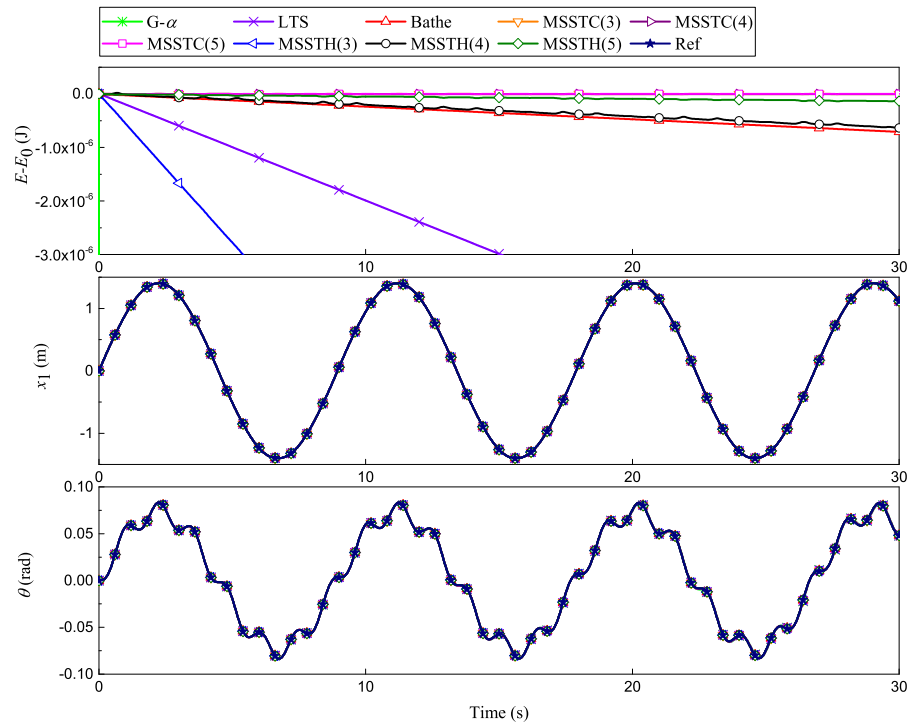
MSSTH(4) is not recommended for these problems due to its poorer stability.

*N-degree-of-freedom mass-spring system* The  $N$ -degree-of-freedom mass-spring system [23], shown in Fig. 24, is considered to check the computational efficiency. The system parameters are set as

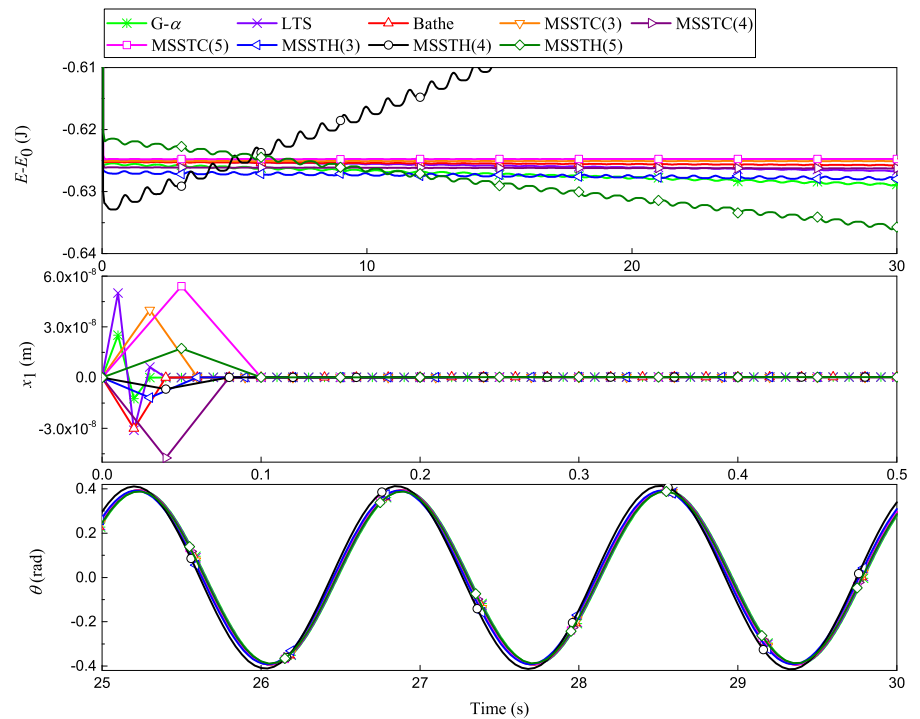
$$m_i = 1 \text{ kg}, f_i = \sin t \text{ N}, i = 1, 2, \dots, N \quad (37a)$$

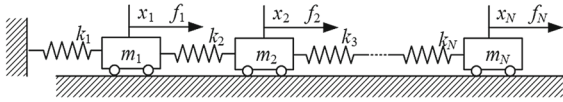
$$k_i = \begin{cases} 10^5 \text{ N/m}, & i = 1 \\ 10^5 \left[ 1 - 2(x_i - x_{i-1})^2 \right] \text{ N/m}, & 2 \leq i \leq N \end{cases} \quad (37b)$$

**Fig. 22** Numerical results of the slider-pendulum model ( $k = 1 \text{ N/m}$ )



**Fig. 23** Numerical results of the slider-pendulum model ( $k = 10^{10} \text{ N/m}$ )





**Fig. 24** Mass-spring model

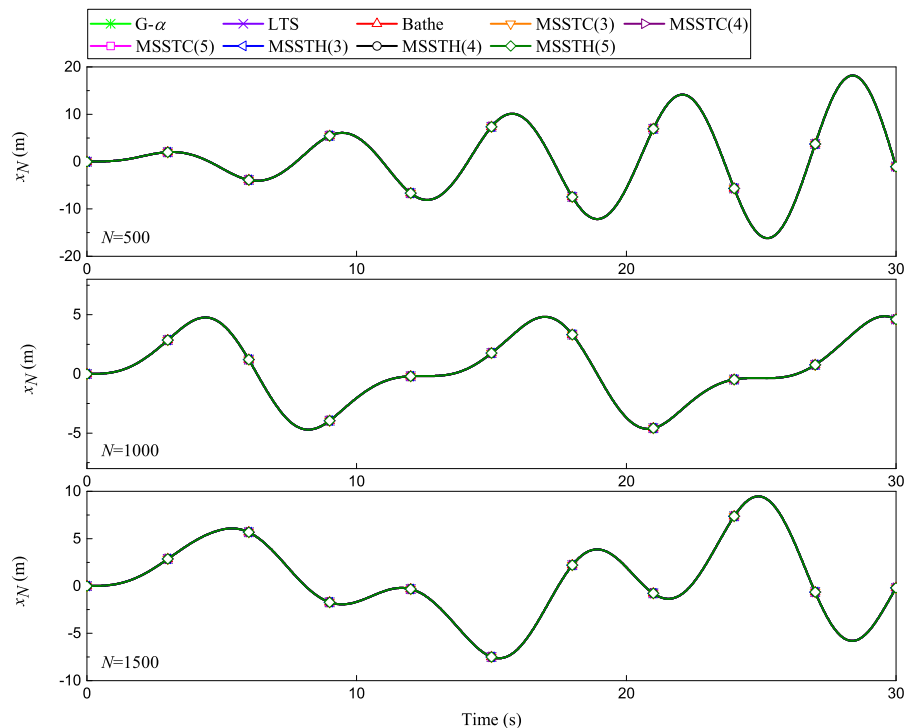
With zero initial conditions, three cases,  $N = 500, 1000$  and  $1500$ , are simulated by these methods using  $h/n = 0.01$  s. Figure 25 shows the numerical solutions of  $x_N$ . It follows that with the step size, all methods can provide reliable results. The CPU time and total number of iterations required by these methods in the simulation of  $[0, 30$  s] are summarized in Table 6. With  $h/n = 0.01$  s, these methods need to proceed 3000 steps (sub-steps for the composite methods) in the whole simulation. Table 6 shows that in addition to MSSTH(4,5), other methods only require one iteration per step/sub-step, so their computational costs are almost equal to each other. One can also see that the required CPU time is approximately proportional to the number of iterations. MSSTH(4,5), especially MSSTH(4), take slightly longer time than other methods.

To check the generality of this conclusion, the required total number of iterations in the above spring-

pendulum and slider-pendulum examples are also listed in Table 7. In the two examples,  $h/n = 0.01$  s is adopted, and the simulation of  $[0, 30$  s] is also performed. The results indicate that the total numbers of iterations required by the second-order methods are very close in all cases. Although the higher-order methods need more iterations sometimes, the increased numbers, especially in MSSTH(3,5), are not very large in most cases. Therefore, it is reasonable to say that these methods with the same  $h/n$  have similar efficiency for nonlinear problems, and the above comparisons in terms of properties are conducted under the close computational costs.

Overall, the numerical examples in this section demonstrate that when applied to nonlinear problems, the proposed methods can still take advantage of their properties, including the energy-conserving characteristic of MSSTC( $n$ ), the high-accuracy of MSSTH( $n$ ), and the strong dissipation ability of both sub-families. However, MSSTH( $n$ ) shows reduced order and energy instability in some examples. From the presented solutions, MSSTH(5) is more recommended in the higher-order sub-family because of its high accuracy and robust stability, whereas MSSTH(4) is not so prefer-

**Fig. 25** Computed  $x_N$  of the mass-spring model



**Table 6** CPU time and total number of iterations required by these methods in the mass-spring example

Method	$N = 500$		$N = 1000$		$N = 1500$	
	CPU time (s)	Number of iterations	CPU time (s)	Number of iterations	CPU time (s)	Number of iterations
G- $\alpha$	34.8852	3000	155.0207	3000	364.3084	3000
LTS	34.1053	3000	164.0188	3000	361.6325	3000
Bathe	33.6191	3000	162.6447	3000	367.3689	3000
MSSTC(3)	34.2762	3000	164.4059	3000	363.7048	3000
MSSTC(4)	32.6436	3000	164.3387	3000	365.8208	3000
MSSTC(5)	33.3455	3000	162.9475	3000	361.8247	3000
MSSTH(3)	38.1994	3000	162.6028	3000	360.1350	3000
MSSTH(4)	52.6115	3988	165.4999	3012	383.2466	3306
MSSTH(5)	47.8183	3289	163.5020	3000	369.4852	3008

**Table 7** Total number of iterations required by these methods in the spring-pendulum and slider-pendulum example

Method	Spring-pendulum example				Slider-pendulum example	
	$f(r) = kr$	$f(r) = kr^3$	$f(r) = k \tanh r$	$f(r) = kr$ (Stiff case)	Compliant case	Stiff case
G- $\alpha$	5999	5999	5997	5925	3161	5980
LTS	5999	6000	5997	5923	3164	5978
Bathe	5998	5997	5997	5902	3285	6001
MSSTC(3)	5999	5998	5998	5890	3000	5978
MSSTC(4)	5999	5998	6000	5893	3000	6003
MSSTC(5)	5999	5999	5997	5891	3000	5992
MSSTH(3)	6000	6001	6000	5989	5751	6007
MSSTH(4)	8405	8286	8355	6006	5973	6663
MSSTH(5)	6704	6583	6676	5998	5828	6139

able, since it shows energy-instability and needs more iterations in some examples.

## 6 Conclusions

In this work, the  $n$ -sub-step composite method ( $n \geq 2$ ), which employs the trapezoidal rule in the first  $n - 1$  sub-steps and a general formula in the last one, is discussed. By optimizing the parameters, the two sub-families, named MSSTC( $n$ ) and MSSTH( $n$ ), are developed, respectively for the energy-conserving and high-accuracy purposes. From linear analysis, MSSTC( $n$ ) and MSSTH( $n$ ) are second-order and  $n$ th-order accurate, respectively, and they can both achieve unconditional stability with controllable algorithmic dissipation. In MSSTC( $n$ ), the purpose of energy-conserving is realized by maximizing the spectral radius in the low-frequency range.

A general approach of parameter optimization, suitable for all schemes with  $n \geq 2$ , is proposed; in this work, the cases  $n = 2, 3, 4, 5$  are discussed in detail. When  $n = 2$ , both sub-families reduce to the  $\rho_\infty$ -Bathe method. As  $n$  increases, MSSTC( $n$ ) shows higher amplitude and period accuracy; its amplitude accuracy is even higher than that of the  $(2n - 1)$ th-order Padé( $n$ ) approximation. MSSTH(3,4,5) exhibits lower period errors than the second-order methods, but their dissipation errors are larger.

The proposed methods are checked on several illustrative examples. The numerical results are mostly consistent with the conclusions from linear analysis. That is, MSSTC( $n$ ) can conserve the energy corresponding to the low-frequency content, and MSSTH( $n$ ) shows higher-order convergence rate for linear and nonlinear equations. However, in the nonlinear examples, some unexpected situations, such as order reduction



and energy instability, emerged in MSSTH( $n$ ). In this sub-family, MSSTH(5) is more recommended thanks to its high-accuracy and robust stability, and MSSTH(4) is not so preferable, since it shows energy instability and lower efficiency in these examples. However, these conclusions about nonlinear problems are obtained from the existing numerical results. The theoretical analysis is still desired in the future.

**Acknowledgements** The first and second authors acknowledge the financial support by the China Scholarship Council.

**Funding** Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

### Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Bank, R.E., Coughran, W.M., Fichtner, W., Grosse, E.H., Rose, D.J., Smith, R.K.: Transient simulation of silicon devices and circuits. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **4**(4), 436–451 (1985)
- Bathe, K.J.: Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Comput. Struct.* **85**(7–8), 437–445 (2007)
- Bathe, K.J., Baig, M.M.I.: On a composite implicit time integration procedure for nonlinear dynamics. *Comput. Struct.* **83**(31–32), 2513–2524 (2005)
- Bathe, K.J., Noh, G.: Insight into an implicit time integration scheme for structural dynamics. *Comput. Struct.* **98**, 1–6 (2012)
- Belytschko, T., Schoeberle, D.: On the unconditional stability of an implicit algorithm for nonlinear structural dynamics. *J. Appl. Mech.* (1975). <https://doi.org/10.1115/1.3423721>
- Butcher, J.C.: Implicit Runge–Kutta processes. *Math. Comput.* **18**(85), 50–64 (1964)
- Butcher, J.C.: *Numerical Methods for Ordinary Differential Equations*. John Wiley, New Jersey (2016)
- Chandra, Y., Zhou, Y., Stanculescu, I., Eason, T., Spottswood, S.: A robust composite time integration scheme for snap-through problems. *Comput. Mech.* **55**(5), 1041–1056 (2015)
- Chung, J., Hulbert, G.: A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- $\alpha$  method. *J. Appl. Mech.* **60**, 371–375 (1993)
- Dahlquist, G.G.: A special stability problem for linear multistep methods. *BIT Numer. Math.* **3**(1), 27–43 (1963)
- Dong, S.: BDF-like methods for nonlinear dynamic analysis. *J. Comput. Phys.* **229**(8), 3019–3045 (2010)
- Fung, T.: Weighting parameters for unconditionally stable higher-order accurate time step integration algorithms. part 1—first-order equations. *Int. J. Numer. Methods Eng.* **45**(8), 941–970 (1999)
- Fung, T.: Weighting parameters for unconditionally stable higher-order accurate time step integration algorithms. Part 2—second-order equations. *Int. J. Numer. Methods Eng.* **45**(8), 971–1006 (1999)
- Fung, T.: Solving initial value problems by differential quadrature method. part 1: first-order equations. *Int. J. Numer. Methods Eng.* **50**(6), 1411–1427 (2001)
- Fung, T.: Solving initial value problems by differential quadrature method. part 2: second-and higher-order equations. *Int. J. Numer. Methods Eng.* **50**(6), 1429–1454 (2001)
- Gear, C.W.: *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall PTR, New Jersey (1971)
- Hilber, H.M., Hughes, T.J., Taylor, R.L.: Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthq. Eng. Struct. Dyn.* **5**(3), 283–292 (1977)
- Ji, Y., Xing, Y.: An optimized three-sub-step composite time integration method with controllable numerical dissipation. *Comput. Struct.* **231**, 106210 (2020)
- Kennedy, C.A., Carpenter, M.H.: Diagonally implicit Runge–Kutta methods for stiff ODEs. *Appl. Numer. Math.* **146**, 221–244 (2019)
- Kim, W., Choi, S.Y.: An improved implicit time integration algorithm: the generalized composite time integration algorithm. *Comput. Struct.* **196**, 341–354 (2018)
- Kim, W., Reddy, J.: An improved time integration algorithm: a collocation time finite element approach. *Int. J. Struct. Stab. Dyn.* **17**(02), 1750024 (2017)
- Li, J., Yu, K., He, H.: A second-order accurate three sub-step composite algorithm for structural dynamics. *Appl. Math. Model.* **77**, 1391–1412 (2020)
- Li, J., Yu, K., Li, X.: A novel family of controllably dissipative composite integration algorithms for structural dynamic analysis. *Nonlinear Dyn.* **96**(4), 2475–2507 (2019)
- Masarati, P., Lanz, M., Mantegazza, P.: Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dynamics applications. In: *Xvi Congresso Nazionale AIDAA*, pp. 1–10 (2001)
- Newmark, N.M.: A method of computation for structural dynamics. *J. Eng. Mech. Div.* **85**(3), 67–94 (1959)
- Noh, G., Bathe, K.J.: The Bathe time integration method with controllable spectral radius: the  $\rho_\infty$ -Bathe method. *Comput. Struct.* **212**, 299–310 (2019)

27. Noh, G., Ham, S., Bathe, K.J.: Performance of an implicit time integration scheme in the analysis of wave propagations. *Comput. Struct.* **123**, 93–105 (2013)
28. Tamma, K.K., Har, J., Zhou, X., Shimada, M., Hoiink, A.: An overview and recent advances in vector and scalar formalisms: space/time discretizations in computational dynamics—a unified approach. *Arch. Comput. Methods Eng.* **18**(2), 119–283 (2011)
29. Wood, W., Bossak, M., Zienkiewicz, O.: An alpha modification of Newmark's method. *Int. J. Numer. Methods Eng.* **15**(10), 1562–1566 (1980)
30. Xie, Y., Steven, G.P.: Instability, chaos, and growth and decay of energy of time-stepping schemes for non-linear dynamic equations. *Commun. Numer. Methods Eng.* **10**(5), 393–401 (1994)
31. Xing, Y., Zhang, H., Wang, Z.: Highly precise time integration method for linear structural dynamic analysis. *Int. J. Numer. Methods Eng.* **116**(8), 505–529 (2018)
32. Zhang, H., Xing, Y.: Optimization of a class of composite method for structural dynamics. *Comput. Struct.* **202**, 60–73 (2018)
33. Zhang, J.: A-stable two-step time integration methods with controllable numerical dissipation for structural dynamics. *Int. J. Numer. Methods Eng.* **121**, 54–92 (2020)
34. Zhou, X., Tamma, K.K.: Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *Int. J. Numer. Methods Eng.* **59**(5), 597–668 (2004)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.