Multibody System Dynamics (2022) 54:263–301
https://doi.org/10.1007/s11044-021-09806-9

Performance of implicit A-stable time integration methods for multibody system dynamics

Huimin Zhang[1,2] · Runsen Zhang[1,2] · Andrea Zanoni[2] · Pierangelo Masarati[2]

Received: 30 June 2021 / Accepted: 22 October 2021 / Published online: 18 January 2022

Abstract
This paper illustrates the performance of several representative implicit A-stable time integration methods with algorithmic dissipation for multibody system dynamics, formulated as a set of mixed implicit first-order differential and algebraic equations. The integrators include the linear multi-step methods with two to four steps, the single-step reformulations of the linear multi-step methods, and explicit first-stage, singly diagonally-implicit Runge–Kutta methods. All methods are implemented in the free, general-purpose multibody solver MBDyn. Their formulations and implementation are presented. According to the comparison from linear analysis and numerical experiments, some general conclusions on the selection of integration schemes and their implementation are obtained. Although all of these methods can predict reasonably accurate solutions, the specific advantages that each of them has in different situations are discussed.

Keywords Implicit · A-stability · Time integration methods · Multibody system dynamics

1 Introduction

Multibody system dynamics problems can be typically formulated as a set of Differential-Algebraic Equations (DAEs), often in semi-explicit form. The numerical treatment of DAEs is more challenging than that of Ordinary Differential Equations (ODEs). Typically, two strategies are employed: direct discretization of DAEs and discretization after reformulation [3]. Reformulation usually consists of some sort of index reduction that can convert DAEs into ODEs, and thus allows the problems to be solved using relatively conventional methods. However, this process may be costly, since it may require substantial user intervention and may be convenient only when a substantial reduction in coordinates can be achieved, which is not the case for examples where mechanisms are analyzed made of flexible components, so direct discretization has gained more attention in software development.

Direct discretization poses strict requirements on time integration methods. Explicit integrators, such as the central difference method [13] and the explicit Runge–Kutta methods [10], are difficult to use to solve DAEs directly because they cannot satisfy the algebraic

✉ H. Zhang
huimin.zhang@polimi.it

1 School of Aeronautic Science and Engineering, Beihang University, Beijing 100083, China

2 Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Milano 20156, Italy

constraint equations at the position level. Implicit integrators, including the linear multi-step methods [23, 24], the implicit Runge–Kutta methods [9] and many direct integration methods [30], are required. They are designed to possess good accuracy, A-stability (unconditional stability), and tunable algorithmic dissipation to provide accurate and robust solutions.

Implicit time integrators are briefly reviewed here. In the field of structural dynamics, several single-step single-solve methods, such as the Newmark method [25], the Wilson-θ method [31], the HHT-α method [17] (proposed by Hilber, Hughes and Taylor), the generalized-α method [11, 22], the GSSSS (Generalized Single-Step Single-Solve) method [37], and many others [18, 32], have been developed since the 1950s. Most of these methods have second-order accuracy, A-stability and tunable algorithmic dissipation from linear analysis. They were initially designed to solve second-order ODEs in structural dynamics, and some of their improved formulations can also be used to solve DAEs and general first-order differential equations [2, 8, 20]. Some comparisons between the linear two-step method and single-step methods have already been presented in [34, 36] and are not reproduced here. Therefore, these single-step methods are not considered further in this work.

In the class of multi-step methods, the linear two-step method [24], and several backward difference formulas (BDFs) [19, 28], have been efficiently used in multibody system dynamics. The optimal parameters of the linear three- and four-step methods with second-order accuracy, A-stability and tunable algorithmic dissipation were given in [36]. According to Dahlquist's second barrier [12], the linear multi-step methods cannot exceed second-order accuracy to possess A-stability. To eliminate the additional starting procedures of the multi-step methods, their equivalent single-step reformulations, obtained by introducing a few auxiliary variables, have been also proposed in [36]. These multi-step and equivalent single-step integrators are designed for systems of first-order differential equations, and can be generalized to solve second- and higher-order differential equations in a straightforward way.

Another important branch of implicit integrators are the multi-stage methods, represented by the Runge–Kutta family [10]. They evaluate the states at intermediate time points per step, and compute the states at discrete time points using a scheme like the quadrature formula. For solving DAEs, the stiffly-accurate Runge–Kutta methods without the quadrature step are more practical, because the constraints are satisfied at the final stage, but may not be satisfied after implementing the quadrature formula. Multi-stage methods can be designed to have higher-order accuracy and A-stability simultaneously, as in [1, 6, 21, 35]. Considering the computational cost, the singly diagonally-implicit Runge–Kutta methods [27], which perform the computation of each stage in sequence, are more convenient and recommended. Consequently, a few recently proposed second- and higher-order stiffly-accurate, singly diagonally-implicit Runge-Kutta methods with explicit first-stage [21, 35] are employed in this work.

The purpose of this work is to present a comparative study of several representative implicit, A-stable and algorithmically dissipative time integration methods for multibody system dynamics. The time integration methods employed include the linear multi-step methods [23, 24, 36], their equivalent single-step methods [36], and stiffly-accurate, explicit first-stage, singly diagonally-implicit Runge–Kutta (ESDIRK) methods [21, 35]. These methods are implemented in the free general-purpose multibody dynamics analysis software MBDyn[1] [24]. Their formulation is presented in Sect. 2, whereas Sect. 3 describes their implementation in MBDyn. Considering a linear problem, the accuracy, algorithmic dissipation

[1] https://www.mbdyn.org.

and dispersion properties of these methods are discussed in Sect. 4. These methods are then applied to solve of some benchmark multibody dynamics problems in Sect. 5. By comparing the numerical results, several general conclusions on the implementation and selection of time integration methods are finally summarized in Sect. 6.

2 Formulation

Initial-value problems in MBDyn are formulated as a set of implicit first-order DAEs, whose general form is

$$\boldsymbol{r}\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, t\right) = \boldsymbol{0}, \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \tag{1}$$

where $\boldsymbol{y}$ collects the differential and algebraic variables, and the overdot indicates the derivative with respect to time t. The initial condition $\boldsymbol{y}_0$ is given, and $\dot{\boldsymbol{y}}_0$ needs to be solved from $\boldsymbol{r}(\boldsymbol{y}_0, \dot{\boldsymbol{y}}_0, t_0) = \boldsymbol{0}$. For a detailed problem description, the reader should refer to [24].

Problems associated with constrained dynamics, as discussed later in Sect. 5, are formulated as Index-3 DAEs, by directly enforcing the holonomic kinematic constraints on the position level through algebraic relations between the coordinates of the problem using Lagrange multipliers. No elaborated scaling strategy is used, except for scaling the Jacobian matrix contribution by the inverse of $\Delta t b_0$, the coefficient that multiplies the derivative of the state at the current time step in implicit numerical schemes, as proposed in [7]. No hidden constraints, e.g., on the velocity level, are considered.

2.1 Linear multi-step methods

The linear multi-step methods can be expressed as

$$\boldsymbol{y}_k = \sum_{j=1}^{r} \alpha_j \boldsymbol{y}_{k-j} + \Delta t \sum_{j=0}^{r} \beta_j \dot{\boldsymbol{y}}_{k-j} \tag{2}$$

where $\boldsymbol{y}_k$ and $\dot{\boldsymbol{y}}_k$ represent the numerical solution at time t_k, $\Delta t = t_k - t_{k-1}$ is the time step size, α_j and β_j are control parameters of the method. Considering the conditions of second-order accuracy, A-stability, and tunable algorithmic dissipation, the optimal parameters of the linear two-, three-, and four-step methods, referred to as LMS2, LMS3 and LMS4, can be found in [36]. They are controlled by a single parameter $\rho_\infty \in [0, 1]$, to adjust the degree of algorithmic dissipation. The algorithmic dissipation becomes stronger as ρ_∞ decreases. When $\rho_\infty = 0$, they reduce to a second-order BDF. In [24], the parameters of LMS2 used in variable-time-step cases were given, while so far LMS3 and LMS4 were previously formulated only for fixed time steps. The initialization of the integration procedure of the multi-step methods, is performed employing the trapezoidal rule for the first few steps

$$\boldsymbol{y}_k = \boldsymbol{y}_{k-1} + \frac{\Delta t}{2}\left(\dot{\boldsymbol{y}}_k + \dot{\boldsymbol{y}}_{k-1}\right) \tag{3}$$

which is A-stable and second-order accurate.

2.2 Equivalent single-step methods

The single-step method proposed in [36], which can be used as an equivalent alternative of the linear multi-step method, has the form

$$\boldsymbol{y}_k = \boldsymbol{y}_{k-1} + \Delta t((1-\gamma_0)\dot{\boldsymbol{y}}_{k-1}^{r-1} + \gamma_0 \dot{\boldsymbol{y}}_k^{r-1}) \tag{4a}$$

$$(1-\gamma_1)\dot{\boldsymbol{y}}_{k-1}^{r-1} + \gamma_1 \dot{\boldsymbol{y}}_k^{r-1} = (1-\gamma_2)\dot{\boldsymbol{y}}_{k-1}^{r-2} + \gamma_2 \dot{\boldsymbol{y}}_k^{r-2} \tag{4b}$$

$$(1-\gamma_3)\dot{\boldsymbol{y}}_{k-1}^{r-2} + \gamma_3 \dot{\boldsymbol{y}}_k^{r-2} = (1-\gamma_4)\dot{\boldsymbol{y}}_{k-1}^{r-3} + \gamma_4 \dot{\boldsymbol{y}}_k^{r-3} \tag{4c}$$

$$\cdots \tag{4d}$$

$$(1-\gamma_{2r-3})\dot{\boldsymbol{y}}_{k-1}^{1} + \gamma_{2r-3} \dot{\boldsymbol{y}}_k^{1} = (1-\gamma_{2r-2})\dot{\boldsymbol{y}}_{k-1} + \gamma_{2r-2}\dot{\boldsymbol{y}}_k \tag{4e}$$

where $\dot{\boldsymbol{y}}_k^j (j = 1, 2, \ldots, r-1)$ are auxiliary variables, and γ_i are control parameters. At the beginning, $\dot{\boldsymbol{y}}_0^1 = \dot{\boldsymbol{y}}_0^2 = \cdots = \dot{\boldsymbol{y}}_k^{r-1} = \dot{\boldsymbol{y}}_0$ is used. The parameters of the single-step methods equivalent to LMS2, LMS3 and LMS4, referred to as SS2, SS3 and SS4, respectively, were also given in [36]. SS3 and SS4 have complex parameters, so the auxiliary variables they produce may be complex numbers, but the state variables are real, as demonstrated in [36].

2.3 Explicit first-stage, singly diagonally-implicit Runge–Kutta (ESDIRK) methods

The stiffly-accurate s-stage ESDIRK is represented using Butcher's tableau [10], as

$$\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & \ldots & 0 & 0 \\
c_2 & a_{21} & \gamma & 0 & \ldots & 0 & 0 \\
c_3 & a_{31} & a_{32} & \gamma & \ddots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
c_{s-1} & a_{s-1,1} & a_{s-1,2} & a_{s-1,3} & \ldots & \gamma & 0 \\
1 & b_1 & b_2 & b_3 & \ldots & b_{s-1} & \gamma \\
\hline
 & b_1 & b_2 & b_3 & \ldots & b_{s-1} & \gamma
\end{array}$$

where γ, a_{ij}, c_i, b_i are control parameters. Considering the implicit first-order equation (1), the intermediate stages are formulated as

$$\boldsymbol{y}_{k-1+c_i} = \boldsymbol{y}_{k-1} + \Delta t \left(\sum_{j=1}^{i-1} a_{ij} \dot{\boldsymbol{y}}_{k-1+c_j} + \gamma \dot{\boldsymbol{y}}_{k-1+c_i} \right) \tag{5}$$

$$i = 2, 3, 4, \ldots, s-1$$

The last stage is

$$\boldsymbol{y}_k = \boldsymbol{y}_{k-1} + \Delta t \left(\sum_{j=1}^{s-1} b_j \dot{\boldsymbol{y}}_{k-1+c_j} + \gamma \dot{\boldsymbol{y}}_k \right) \tag{6}$$

where $c_1 = 0$ and $c_s = 1$. The computation of each stage can be implemented in sequence. Except for the explicit first stage, every other stage uses an implicit single-step or multi-step formula. They share the same effective stiffness matrix for linear analysis and the same form

of Jacobian matrix in nonlinear iterations. In the procedure, the state variables at the time points are given, but those at the intermediate stages are not output. Therefore, the multi-stage methods are essentially single-step schemes, which only use the information from the last step to update the current one.

The two-sub-step ρ_∞-Bathe method [26], which can be seen as a 3-stage ESDIRK with second-order accuracy, A-stability and tunable algorithmic dissipation, is presented. Its parameters are

$$\gamma = \begin{cases} \dfrac{2-\sqrt{2(1+\rho_\infty)}}{2(1-\rho_\infty)}, & \rho_\infty \in [0,1) \\ \dfrac{1}{4}, & \rho_\infty = 1 \end{cases} \qquad (7)$$

$$b_1 = -\frac{4\gamma^2 - 6\gamma + 1}{4\gamma}, \qquad b_2 = \frac{1-2\gamma}{4\gamma}$$

The multi-sub-step methods MSSTC(n) and MSSTH(n), as $(n+1)$-stage ESDIRKs proposed in [35], are also employed. MSSTC(n) is designed to have second-order accuracy, A-stability, tunable algorithmic dissipation, preserving low-frequency dynamics as much as possible. It employs the trapezoidal rule from the second to the nth stage, and a general formula in the last one. The optimal parameters of MSSTC(3, 4, 5) were given in [35].

MSSTH(n) is designed to have nth-order accuracy, A-stability, and tunable algorithmic dissipation. However, since only linear analysis was considered in [35], its parameters are modified here to satisfy the overall-order and stage-order conditions for Runge–Kutta methods [10, 16, 21], without changing the linear characteristics. The design of the modified MSSTH(3, 4, 5) considers the following conditions:

- Each stage, from the second-one, has at least second-order accuracy;
- The overall accuracy is nth-order, and on this basis, the local truncation error is minimized;
- A-stability and tunable algorithmic dissipation for linear analysis.

Under the premise of A-stability, MSSTH(n) is designed to achieve the highest possible accuracy for a given ρ_∞. The details of the parameter selection are shown in the Appendix.

Besides, several stiffly-accurate ESDIRKs developed in [21], including the third-order 4-stage ESDIRK3(2)4L[2]SA, the third-order 5-stage ESDIRK3(2)5L[2]SA, and the fourth-order 6-stage ESDIRK4(3)6L[2]SA$_2$, are also employed. These methods are designed to be L-stable ($\rho_\infty = 0$), and their internal stages are also designed to be L-stable whenever possible. Their parameters can be found in [21].

3 Implementation

The detailed problem description, solution phases and implementation structure of MBDyn were presented in [24]. Time integrators are defined in the class `ImplicitStepIntegrator`. The multi-step, single-step, and multi-stage integrators are constructed using the template classes `tplStepNIntegrator`, `tplSingleStepIntegrator` and `tplStageNIntegrator`, respectively. They provide an `Advance()` method to perform a complete step. For multi-stage integrators, the operations of all stages are encapsulated in the method `Advance()`, so the solutions in the internal stages are hidden.

With the initial condition at t_0, the multi-step and single-step integrators sequentially calculate the state variables at t_1, t_2, t_3, $\cdots$, t_k, $\cdots$, and the ESDIRKs need to solve the results at t_{0+c_2}, t_{0+c_3}, $\cdots$, $t_{0+c_{s-1}}$, t_1, t_{1+c_2}, t_{1+c_3}, $\cdots$, $t_{1+c_{s-1}}$, t_2, $\cdots$, t_k, $\cdots$ in turn. Note that the explicit first stage of ESDIRKs does not require any calculation. All methods show a very similar structure at each time point; the difference lies in the parameters and the number of previous time points used. The single-step methods, must additionally solve and store the auxiliary variables. Therefore, the implementation at a certain discrete time point, t_N, which is used to represent all time points, including those at the end of each step t_k and the internal time points of the ESDIRKs, is explained uniformly in this section. For multi-step and single-step methods, the information at t_{N-1}, t_{N-2}, $\cdots$, used at the current time point t_N are the states of previous steps. For ESDIRKs, the information of t_{N-1}, t_{N-2}, $\cdots$, are the states of the previously solved time points, which can be the last step and last stages.

At the discrete time point t_N, the state variables $\boldsymbol{y}_N$ and $\dot{\boldsymbol{y}}_N$ are obtained by solving

$$\boldsymbol{r}\left(\boldsymbol{y}_N, \dot{\boldsymbol{y}}_N, t_N\right) = \boldsymbol{0} \tag{8a}$$

$$\boldsymbol{y}_N = \boldsymbol{f}\left(\boldsymbol{y}_{N-1}, \boldsymbol{y}_{N-2}, \ldots, \dot{\boldsymbol{y}}_N, \dot{\boldsymbol{y}}_{N-1}, \dot{\boldsymbol{y}}_{N-2}, \ldots, \dot{\boldsymbol{y}}^1_{N-1}, \dot{\boldsymbol{y}}^2_{N-1}, \ldots, \dot{\boldsymbol{y}}^r_{N-1}\right) \tag{8b}$$

Here Eq. (8b) represents the integration scheme. It is Eq. (2) for multi-step integrators, and Eq. (5) or Eq. (6) for the ESDIRKs. For single-step methods, it needs to be reorganized from Eqs. (4a)–(4e). SS2 uses

$$\boldsymbol{y}_N = \boldsymbol{y}_{N-1} + \Delta t \left(\frac{\gamma_0\gamma_2}{\gamma_1} \dot{\boldsymbol{y}}_N + \frac{\gamma_0(1-\gamma_2)}{\gamma_1} \dot{\boldsymbol{y}}_{N-1} + \frac{\gamma_1-\gamma_0}{\gamma_1} \dot{\boldsymbol{y}}^1_{N-1} \right) \tag{9}$$

SS3 uses

$$\begin{aligned} \boldsymbol{y}_N &= \boldsymbol{y}_{N-1} \\ &+ \Delta t \left(\frac{\gamma_0\gamma_2\gamma_4}{\gamma_1\gamma_3} \dot{\boldsymbol{y}}_N + \frac{\gamma_0\gamma_2(1-\gamma_4)}{\gamma_1\gamma_3} \dot{\boldsymbol{y}}_{N-1} + \frac{\gamma_0(\gamma_3-\gamma_2)}{\gamma_1\gamma_3} \dot{\boldsymbol{y}}^1_{N-1} + \frac{\gamma_1-\gamma_0}{\gamma_1} \dot{\boldsymbol{y}}^2_{k-1} \right) \end{aligned} \tag{10}$$

SS4 uses

$$\begin{aligned} \boldsymbol{y}_N &= \boldsymbol{y}_{N-1} \\ &+ \Delta t \left(\frac{\gamma_0\gamma_2\gamma_4\gamma_6}{\gamma_1\gamma_3\gamma_5} \dot{\boldsymbol{y}}_N + \frac{\gamma_0\gamma_2\gamma_4(1-\gamma_6)}{\gamma_1\gamma_3\gamma_5} \dot{\boldsymbol{y}}_{N-1} + \right. \\ &\left. \frac{\gamma_0\gamma_2(\gamma_5-\gamma_4)}{\gamma_1\gamma_3\gamma_5} \dot{\boldsymbol{y}}^1_{N-1} + \frac{\gamma_0(\gamma_3-\gamma_2)}{\gamma_1\gamma_3} \dot{\boldsymbol{y}}^2_{k-1} + \frac{\gamma_1-\gamma_0}{\gamma_1} \dot{\boldsymbol{y}}^3_{k-1} \right) \end{aligned} \tag{11}$$

3.1 Prediction

A predictor-corrector approach [24] is used to solve Eqs. (8a)–(8b) in MBDyn. Prediction and correction are two separate, independent, and consecutive phases. The objective of the prediction phase is to determine a tentative value for the solution to start the correction. Here $\dot{\boldsymbol{y}}^{(0)}_N$ is used to represent the predicted value, and then $\boldsymbol{y}^{(0)}_N$ can be obtained directly by the integration scheme. The scheme for prediction has minimal effect on the accuracy of the solutions, but it may affect the number of iterations required, as shown in Sect. 5. The closer the predicted value is to the final solution, the fewer iterations are required. Consequently, the criterion for choosing the prediction scheme here is that it should not employ extra

information that was not used in the integration scheme, and that it should have similar accuracy order to the integration scheme.

For the multi-step integrators, $\dot{\boldsymbol{y}}_N^{(0)}$ is predicted by a second-order scheme, as

$$\dot{\boldsymbol{y}}_N^{(0)} = \frac{1}{t_N - t_{N-1}}(m_0 \boldsymbol{y}_{N-1} + m_1 \boldsymbol{y}_{N-2}) + n_0 \dot{\boldsymbol{y}}_{N-1} + n_1 \dot{\boldsymbol{y}}_{N-2} \tag{12}$$

Its local truncation error is defined as

$$\sigma = \dot{\boldsymbol{y}}(t_N) - n_0 \dot{\boldsymbol{y}}(t_{N-1}) - n_1 \dot{\boldsymbol{y}}(t_{N-2}) - \frac{1}{t_N - t_{N-1}}\left(m_0 \boldsymbol{y}(t_{N-1}) + m_1 \boldsymbol{y}(t_{N-2})\right) \tag{13}$$

Expanding the right-hand side at t_N by Taylor's theorem and letting the local truncation error satisfy $\sigma = O((t_N - t_{N-1})^3)$, i.e., second-order accuracy, yields

$$\begin{aligned} \alpha &= \frac{t_N - t_{N-1}}{t_{N-1} - t_{N-2}} \\ m_0 &= -6\alpha^2(1+\alpha), & m_1 &= -m_0 \\ n_0 &= (1+\alpha)(1+3\alpha), & n_1 &= \alpha(2+3\alpha) \end{aligned} \tag{14}$$

For the single-step integrators, the integration schemes use only the states of the last step. So to make them truly single-step, the constant prediction is used, as

$$\dot{\boldsymbol{y}}_N^{(0)} = \dot{\boldsymbol{y}}_{N-1} \tag{15}$$

For the ESDIRKs, if t_N is in the second-stage, using only the states of the last time point, the constant prediction in Eq. (15) is employed. If t_N is in the third to $(s-1)$th stage, since the integration scheme in these stages are all second-order accurate, the second-order prediction in Eq. (12) is used. Note that in these cases t_N is the current time and t_{N-1} is the time of the last stage, so $t_N - t_{N-1}$ in Eq. (12) is not the time step size Δt. The accuracy of the last stage has the same order as the overall accuracy, so the second-order ρ_∞-Bathe method, MSSTC(3, 4, 5), as well as the third-order MSSTH(3), ESDIRK3(2)4L[2]SA and ESDIRK3(2)5L[2]SA still use second-order prediction in the last stage, but for the last stage of the fourth-order MSSTH(4), ESDIRK4(3)6L[2]SA$_2$ and the fifth-order MSSTH(5), a fourth-order prediction scheme is employed, as

$$\begin{aligned} \dot{\boldsymbol{y}}_N^{(0)} = \frac{1}{t_N - t_{N-1}}&\left(m_0 \boldsymbol{y}_{N-1} + m_1 \boldsymbol{y}_{N-2} + m_2 \boldsymbol{y}_{N-3}\right) \\ &+ n_0 \dot{\boldsymbol{y}}_{N-1} + n_1 \dot{\boldsymbol{y}}_{N-2} + n_2 \dot{\boldsymbol{y}}_{N-3} \end{aligned} \tag{16}$$

The parameters are determined by making it fourth-order accurate, as

$$\alpha_1 = \frac{t_N - t_{N-1}}{t_{N-1} - t_{N-2}}, \qquad \alpha_2 = \frac{t_N - t_{N-1}}{t_{N-2} - t_{N-3}} \tag{17a}$$

$$\begin{aligned} m_0 = -\frac{2\alpha_1^2(1+\alpha_1)}{(\alpha_1+\alpha_2)^3}&\left(5\alpha_1^3\alpha_2^2 + 8\alpha_1^3\alpha_2 + 10\alpha_1^2\alpha_2^3 + \right. \\ &\left. 3\alpha_1^3 + 25\alpha_1^2\alpha_2^2 + 13\alpha_1^2\alpha_2 + 20\alpha_1\alpha_2^3 + 20\alpha_1\alpha_2^2 + 10\alpha_2^3\right) \end{aligned} \tag{17b}$$

$$m_1 = \frac{2(1+\alpha_1)}{\alpha_1}\left(5\alpha_1^3\alpha_2^2 + 8\alpha_1^3\alpha_2 + 3\alpha_1^3 - 5\alpha_1^2\alpha_2^3 + \right. \tag{17c}$$

$$\alpha_1^2\alpha_2^2 + 4\alpha_1^2\alpha_2 - 7\alpha_1\alpha_2^3 - \alpha_1\alpha_2^2 - 2\alpha_2^3\big)$$

$$m_2 = -m_0 - m_1 \tag{17d}$$

$$n_0 = \frac{(1+\alpha_1)}{(\alpha_1+\alpha_2)^2}\big(5\alpha_1^3\alpha_2^2 + 8\alpha_1^3\alpha_2 + 3\alpha_1^3 + 11\alpha_1^2\alpha_2^2 \tag{17e}$$

$$+10\alpha_1^2\alpha_2 + \alpha_1^2 + 7\alpha_1\alpha_2^2 + 2\alpha_1\alpha_2 + \alpha_2^2\big)$$

$$n_1 = \frac{1}{\alpha_1}\big(5\alpha_1^3\alpha_2^2 + 8\alpha_1^3\alpha_2 + 3\alpha_1^3 + 12\alpha_1^2\alpha_2^2 + \tag{17f}$$

$$12\alpha_1^2\alpha_2 + 2\alpha_1^2 + 9\alpha_1\alpha_2^2 + 4\alpha_1\alpha_2 + 2\alpha_2^2\big)$$

$$n_2 = \frac{\alpha_2^3(1+\alpha_1)}{\alpha_1(\alpha_1+\alpha_2)^2}\big(2\alpha_1 + 2\alpha_2 + 7\alpha_1\alpha_2 + 5\alpha_1^2\alpha_2 + 4\alpha_1^2\big) \tag{17g}$$

In terms of rotations, the orientation and angular velocity of each node used for spatial modelling in MBDyn are stored in the orientation matrix $\mathbf{R}$, and vector $\boldsymbol{\omega}$, respectively. To predict the orientation at t_N, the Cayley–Gibbs–Rodriguez (CGR) orientation parameters are assumed to be zero at the last point t_{N-1}, namely $\boldsymbol{g}_{k-1} \equiv \mathbf{0}$, and those of the other previous steps, if needed, are extracted from the respective orientation matrices relative to that at time t_{N-1}, namely $\boldsymbol{g}_{N-j} = \boldsymbol{g}(\mathbf{R}_{N-j}\mathbf{R}_{N-1}^T)$, $j \geq 2$. This procedure resembles typical approaches to Lie group integration (e.g., [14]), and was inspired by the spatial interpolation of finite rotations on 1D domains, originally formulated for geometrically exact beam analysis.

The CGR parameter derivatives are evaluated accordingly: $\dot{\boldsymbol{g}}_{N-1} \equiv \boldsymbol{\omega}_{N-1}$, since $\boldsymbol{g}_{N-1} \equiv \mathbf{0}$, and $\dot{\boldsymbol{g}}_{N-j} = \mathbf{G}_{N-j}^{-1}\boldsymbol{\omega}_{N-j}$, $j \geq 2$, where the matrix $\mathbf{G}(\boldsymbol{g})$ expresses the transformation from the rotation parameter derivatives to the angular velocity, $\boldsymbol{\omega} = \mathbf{G}\dot{\boldsymbol{g}}$. For the detailed expressions of $\boldsymbol{g}(\mathbf{R})$ and $\mathbf{G}(\boldsymbol{g})$ please refer to [24].

The single-step integrators must additionally prepare the auxiliary variables $\dot{\boldsymbol{g}}_{N-1}^p$ ($1 \leq p \leq r-1$), but they do not have the corresponding CGR parameters $\boldsymbol{g}_{N-1}^p$. To simplify the computation, the approximation $\boldsymbol{g}_{N-1}^p \approx \boldsymbol{g}_{N-1} \equiv \mathbf{0}$ is used, such that $\dot{\boldsymbol{g}}_{N-1}^p \equiv \boldsymbol{\omega}_{N-1}^p$. After obtaining the final solutions at t_{N-1}, $\boldsymbol{\omega}_{N-1}^p$ are updated according to the integration scheme. From Eqs. (4a)–(4e), SS2 uses

$$\boldsymbol{\omega}_{N-1}^1 = \frac{1-\gamma_2}{\gamma_1}\boldsymbol{\omega}_{N-2} + \frac{\gamma_2}{\gamma_1}\boldsymbol{\omega}_{N-1} - \frac{1-\gamma_1}{\gamma_1}\boldsymbol{\omega}_{N-2}^1 \tag{18}$$

SS3 uses

$$\boldsymbol{\omega}_{N-1}^1 = \frac{1-\gamma_4}{\gamma_3}\boldsymbol{\omega}_{N-2} + \frac{\gamma_4}{\gamma_3}\boldsymbol{\omega}_{N-1} - \frac{1-\gamma_3}{\gamma_3}\boldsymbol{\omega}_{N-2}^1 \tag{19a}$$

$$\boldsymbol{\omega}_{N-1}^2 = \frac{1-\gamma_2}{\gamma_1}\boldsymbol{\omega}_{N-2}^1 + \frac{\gamma_2}{\gamma_1}\boldsymbol{\omega}_{N-1}^1 - \frac{1-\gamma_1}{\gamma_1}\boldsymbol{\omega}_{N-2}^2 \tag{19b}$$

SS4 uses

$$\boldsymbol{\omega}_{N-1}^1 = \frac{1-\gamma_6}{\gamma_5}\boldsymbol{\omega}_{N-2} + \frac{\gamma_6}{\gamma_5}\boldsymbol{\omega}_{N-1} - \frac{1-\gamma_5}{\gamma_5}\boldsymbol{\omega}_{N-2}^1 \tag{20a}$$

$$\boldsymbol{\omega}_{N-1}^2 = \frac{1-\gamma_4}{\gamma_3}\boldsymbol{\omega}_{N-2}^1 + \frac{\gamma_4}{\gamma_3}\boldsymbol{\omega}_{N-1}^1 - \frac{1-\gamma_3}{\gamma_3}\boldsymbol{\omega}_{N-2}^2 \tag{20b}$$

$$\boldsymbol{\omega}_{N-1}^3 = \frac{1-\gamma_2}{\gamma_1}\boldsymbol{\omega}_{N-2}^2 + \frac{\gamma_2}{\gamma_1}\boldsymbol{\omega}_{N-1}^2 - \frac{1-\gamma_1}{\gamma_1}\boldsymbol{\omega}_{N-2}^3 \tag{20c}$$

Certainly, other derivatives involved in $\dot{\boldsymbol{y}}_{N-1}$ must be updated in the same way before computing at t_N. The auxiliary variables of SS3 and SS4 may be complex numbers, so their real and imaginary parts are computed and stored separately. Due to the approximation of rotations, these single-step integrators are no longer exactly equivalent to the corresponding multi-step methods, but their solutions in numerical experiments are still very close, as shown in Sect. 5. This indicates that the approximation does not cause any obvious reduction in accuracy.

Having obtained the required $\boldsymbol{g}_{N-j}$, $\dot{\boldsymbol{g}}_{N-j}$ $(j \geq 1)$ and $\boldsymbol{g}'^{p}_{N-1}$ $(1 \leq p \leq r-1)$, the CGR parameters and their derivatives at time t_N, $\boldsymbol{g}^{(0)}_N$ and $\dot{\boldsymbol{g}}^{(0)}_N$, are predicted using the previously illustrated schemes. Then, the predicted orientation matrix and angular velocity are computed as $\mathbf{R}_\Delta = \mathbf{R}(\boldsymbol{g}^{(0)}_N)$, $\mathbf{R}^{(0)}_N = \mathbf{R}_\Delta \mathbf{R}_{N-1}$ and $\mathbf{G}_\Delta = \mathbf{G}(\boldsymbol{g}^{(0)}_N)$, $\boldsymbol{\omega}^{(0)} = \mathbf{G}_\Delta \dot{\boldsymbol{g}}^{(0)}_N$.

Of course, the relative rotation between the involved time steps is assumed limited to avoid the indeterminacies and singularities inherent in all three-parameter parameterizations, specifically those related to the CGR parameters (the magnitude of the relative rotations must be significantly smaller than π). Such an assumption is considered acceptable when the time step of the integration is dictated by accuracy requirements.

3.2 Correction

After the prediction phase, by a Newton-like iteration method, $\boldsymbol{y}_N$ and $\dot{\boldsymbol{y}}_N$ are corrected according to

$$\left[\frac{\partial \boldsymbol{y}_N}{\partial \dot{\boldsymbol{y}}_N}\boldsymbol{r}_{\boldsymbol{y}}\left(\boldsymbol{y}^{(l)}_N, \dot{\boldsymbol{y}}^{(l)}_N, t_N\right) + \boldsymbol{r}_{\dot{\boldsymbol{y}}}\left(\boldsymbol{y}^{(l)}_N, \dot{\boldsymbol{y}}^{(l)}_N, t_N\right)\right]\Delta\dot{\boldsymbol{y}} = -\boldsymbol{r}\left(\boldsymbol{y}^{(l)}_N, \dot{\boldsymbol{y}}^{(l)}_N, t_N\right) \tag{21a}$$

$$\dot{\boldsymbol{y}}^{(l+1)}_N = \dot{\boldsymbol{y}}^{(l)}_N + \Delta\dot{\boldsymbol{y}} \tag{21b}$$

$$\boldsymbol{y}^{(l+1)}_N = \boldsymbol{y}^{(l)}_N + \frac{\partial \boldsymbol{y}_N}{\partial \dot{\boldsymbol{y}}_N}\Delta\dot{\boldsymbol{y}} \tag{21c}$$

where $\boldsymbol{r}_{\boldsymbol{y}}$ and $\boldsymbol{r}_{\dot{\boldsymbol{y}}}$ are the partial derivatives of $\boldsymbol{r}$ with respect to $\boldsymbol{y}$ and $\dot{\boldsymbol{y}}$, respectively; $\partial \boldsymbol{y}_N/\partial \dot{\boldsymbol{y}}_N = \beta_0 \Delta t$ for the multi-step integrators, $\partial \boldsymbol{y}_N/\partial \dot{\boldsymbol{y}}_N = \gamma_0\gamma_2\Delta t/\gamma_1$ for SS2, $\partial \boldsymbol{y}_N/\partial \dot{\boldsymbol{y}}_N = \gamma_0\gamma_2\gamma_4\Delta t/(\gamma_1\gamma_3)$ for SS3, $\partial \boldsymbol{y}_N/\partial \dot{\boldsymbol{y}}_N = \gamma_0\gamma_2\gamma_4\gamma_6\Delta t/(\gamma_1\gamma_3\gamma_5)$ for SS4, and $\partial \boldsymbol{y}_N/\partial \dot{\boldsymbol{y}}_N = \gamma\Delta t$ for the ESDIRKs; l denotes the number of iterations, and $l = 0$ at the initial.

In the correction phase, at each iteration, $\boldsymbol{g}^{(l)}_N$ and $\dot{\boldsymbol{g}}^{(l)}_N$ are obtained through Eqs. (21a)–(21c). The orientation is recast as $\mathbf{R}^{(l)}_N = \mathbf{R}_\Delta \mathbf{R}^{(0)}_N$ with $\mathbf{R}_\Delta = \mathbf{R}(\boldsymbol{g}^{(l)}_N)$. The angular velocity is expressed as $\boldsymbol{\omega}^{(l)}_N = \mathbf{G}_\Delta \dot{\boldsymbol{g}}^{(l)}_N + \mathbf{R}_\Delta \boldsymbol{\omega}^{(0)}_N$ with $\mathbf{G}_\Delta = \mathbf{G}(\boldsymbol{g}^{(l)}_N)$.

The final solution at t_N is obtained when the prescribed convergence condition is satisfied. Then the procedure moves on to the solution for the next time point.

3.3 Generalization and extension

All previously described methods, which are summarized in Fig. 1, are available in MBDyn's public source code repository.[2] Other time integration schemes can be easily added using the provided class templates.

[2] https://public.gitlab.polimi.it/DAER/mbdyn, in the "integrators" development branch, to be merged soon with the main development branch.

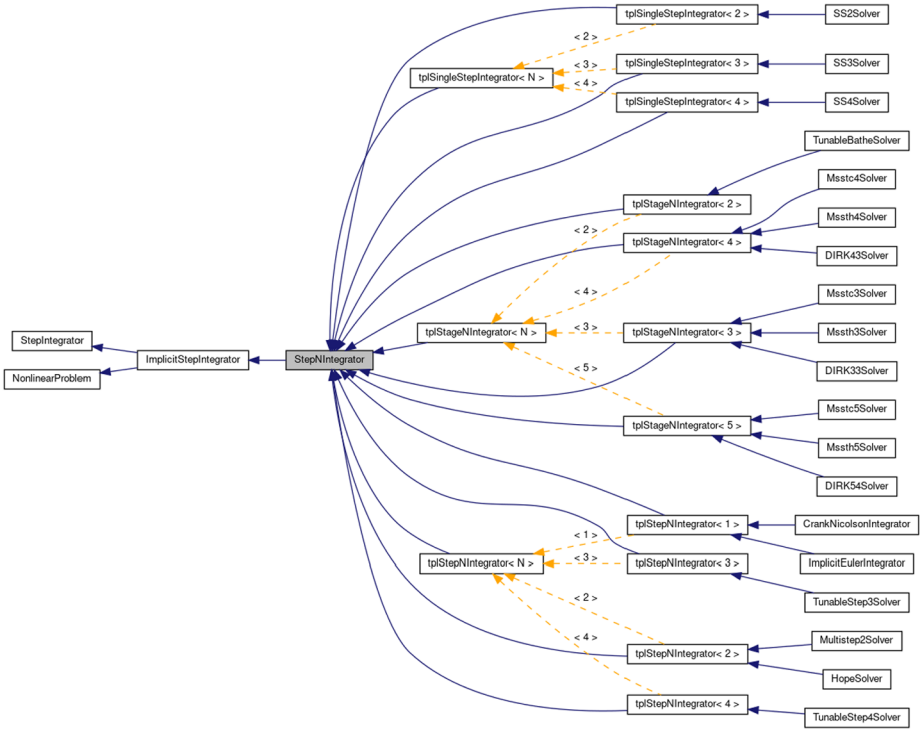

Fig. 1 Inheritance tree of the `StepNIntegrator` class, generated using Doxygen

To define a new integrator that belongs to the families of linear multi-step, equivalent single-step, or multi-stage methods, one simply needs to derive a new class from the corresponding template class, define the number of steps or stages (an integer template value), and overload the virtual methods that provide the coefficients for the prediction of state and derivative.

The implementation of these methods in MBDyn granted the possibility to evaluate their performance when applied to non-trivial, general-purpose multibody system dynamics problems. Some examples taken from the literature are investigated in Sect. 5.

4 Linear analysis

The dissipation and dispersion accuracy as well as the degree of algorithmic dissipation of the employed methods are compared in this section considering the single degree-of-freedom problem $\ddot{x} + \omega^2 x = 0$, an undamped oscillator. The numerical solution of the problem at time t_k can be expressed as

$$x_k = \mathrm{e}^{-\overline{\xi}\overline{\omega}t_k} \left(c_1 \cos(\overline{\omega}_d t_k) + c_2 \sin(\overline{\omega}_d t_k)\right), \quad \overline{\omega}_d = \overline{\omega}\sqrt{1-\overline{\xi}^2} \tag{22}$$

where c_1 and c_2 are constants determined by the initial conditions, and $\overline{\omega}$ and $\overline{\xi}$ are the numerical natural frequency and damping ratio, respectively. Here $\overline{\xi}$ and $(\overline{T} - T)/T = \omega/\overline{\omega} - 1$ are known as the amplitude decay ratio and period elongation ratio, respectively. They are

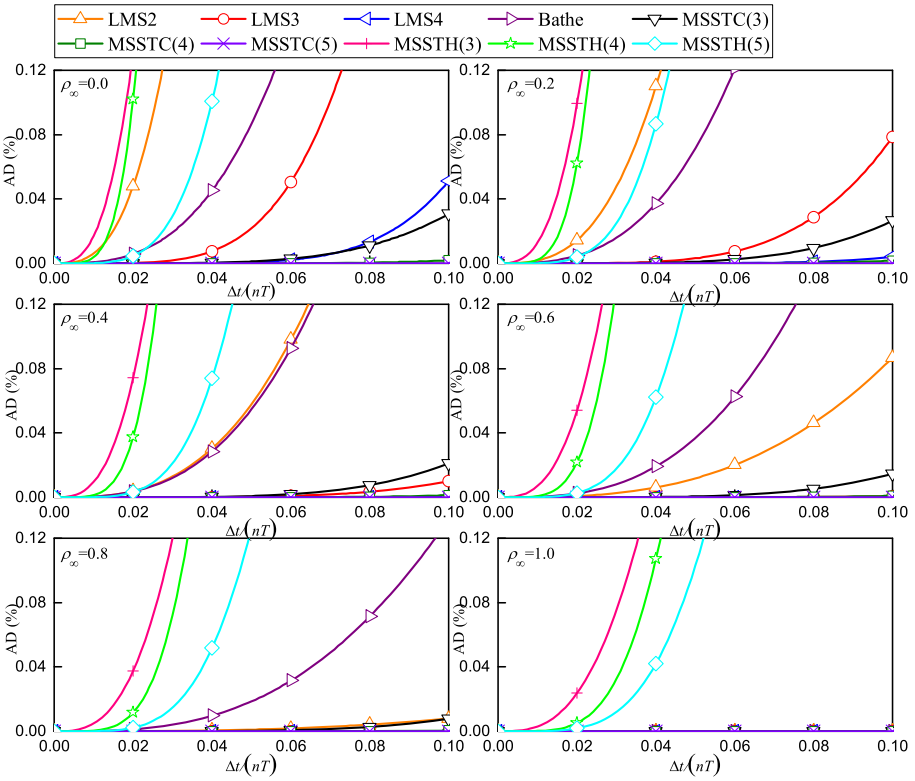

Fig. 2 Percentage amplitude decay of the methods with tunable algorithmic dissipation

typically used to measure the dissipation and dispersion accuracy in the low-frequency domain. They can be obtained from the characteristic roots of the method, as in [37]. Besides, the spectral radius of the method is used to measure the degree of algorithmic dissipation.

Figures 2, 3 and 4 summarize the percentage amplitude decay (AD(%)), percentage period elongation (PE(%)) and spectral radius (ρ) of methods with tunable algorithmic dissipation, respectively. Considering the intrinsic spectral equivalence of the single-step methods and the corresponding multi-step methods, only the results of the multi-step methods are shown. Figures 5, 6 and 7 show AD(%), PE(%) and ρ of the higher-order ESDIRKs with $\rho_\infty = 0.0$. Because the s-stage ESDIRKs perform $s-1$ implicit stages per step, to compare the results under similar computational cost, the abscissa is set as $\Delta t/(nT)$ in the figures, where $n = 1$ for the multi-step methods, and $n = s-1$ for the s-stage ESDIRKs.

The employed methods all have A-stability and can provide algorithmic dissipation. Among them, LMS2, LMS3, LMS4, Bathe, MSSTC(3, 4, 5) are second-order accurate, and other ESDIRKs have higher-order accuracy. From the comparison, one can observe that the second-order methods, especially LMS3, LMS4, MSSTC(3, 4, 5), are really superior in preserving the frequency content. They show very small numerical damping ratios and spectral radii very close to 1 when $\Delta t/(nT) \leq 0.1$.

As the number of steps or stages used increases, both dissipation and dispersion accuracy improve for second-order methods, so LMS4 and MSSTC(5) have better accuracy, and can retain more frequency content. As ρ_∞ grows, these second-order methods also show higher

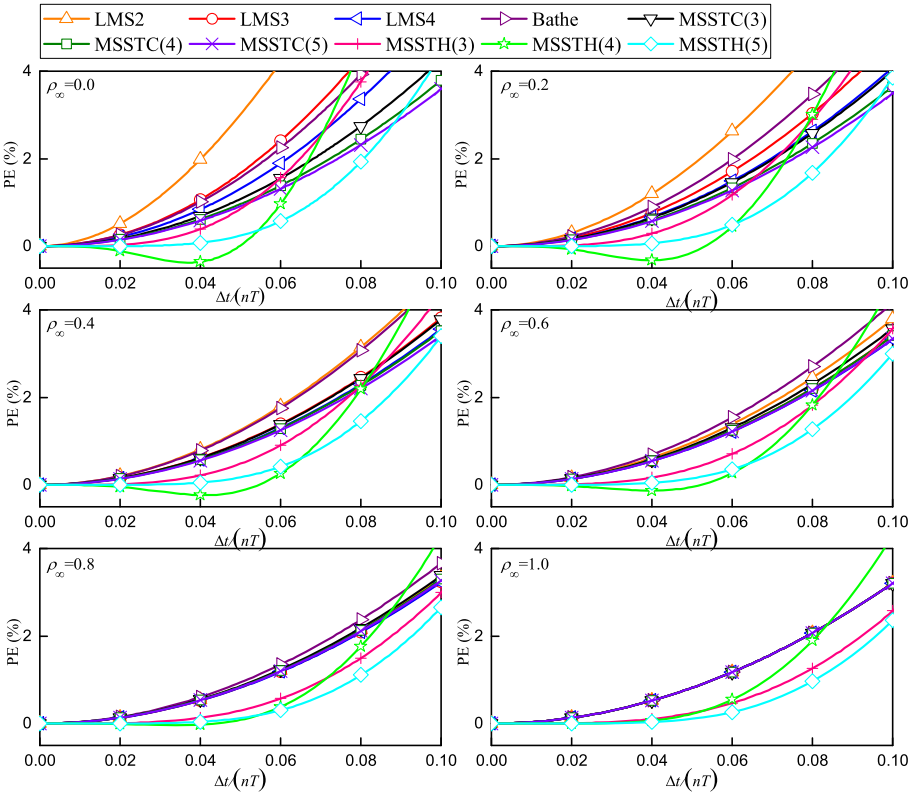

Fig. 3 Percentage period elongation of the methods with tunable algorithmic dissipation

accuracy, and when $\rho_\infty = 1.0$, they all have the same characteristics as the trapezoidal rule. Compared to the multi-step methods, the second-order multi-stage ones show better filtering ability for high-frequency content, since their spectral radii drop to ρ_∞ faster and earlier.

Compared to the second-order methods, the higher-order ESDIRKs exhibit higher dispersion accuracy, which gives them some accuracy advantage in transient simulations. On the other hand, the excessive algorithmic dissipation of MSSTH(3, 4, 5) and ESDIRK3(2)4L[2]SA dissipates most of the frequency content in long-term simulations. With $\rho_\infty = 0.0$, MSSTH(5), ESDIRK3(2)5L[2]SA and ESDIRK4(3)6L[2]SA$_2$ show very small period elongation for $\Delta t/(nT) \leq 0.05$, whereas MSSTH(4) exhibits period shortening. MSSTH(3) with $\rho_\infty = 0.0$ has spectral characteristics almost identical to those of ESDIRK3(2)4L[2]SA. Even though MSSTH(3, 4, 5) have tunable algorithmic dissipation, their accuracy cannot be improved when ρ_∞ increases, and when $\rho_\infty = 1.0$ they exhibit unexpected algorithmic dissipation at intermediate frequencies. For these reasons, a small ρ_∞, like 0.0, is recommended for MSSTH(3, 4, 5) to improve stability of the algorithm.

From the linear analysis, it appears that the second-order methods are very effective in preserving the amplitude, while the higher-order methods have better phase accuracy. Among second-order methods, the use of more steps or stages or a larger ρ_∞ helps to improve the dissipation and dispersion accuracy. Multi-step methods can retain more frequency content than multi-stage ones with the same ρ_∞. Except for ESDIRK4(3)6L[2]SA$_2$

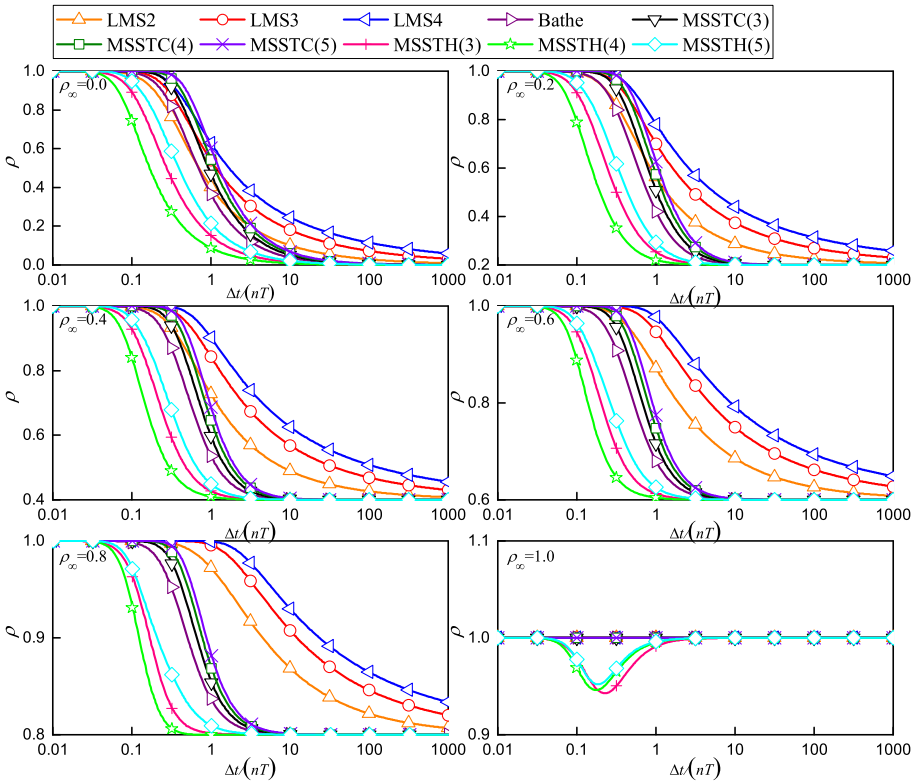

Fig. 4 Spectral radius of the methods with tunable algorithmic dissipation

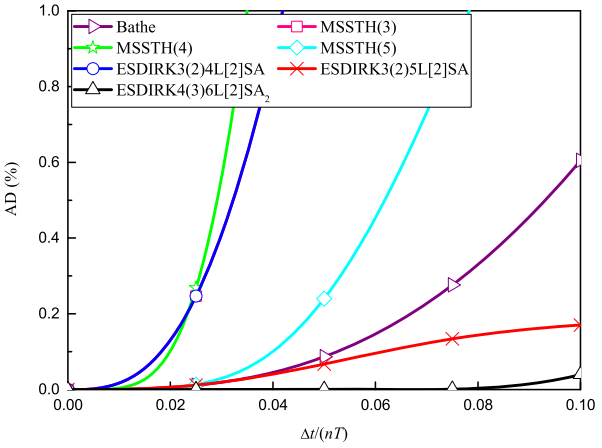

Fig. 5 Percentage amplitude decay of the higher-order methods with $\rho_\infty = 0.0$

and ESDIRK3(2)5L[2]SA, the employed other higher-order methods show excessive algorithmic dissipation in the low-frequency domain.

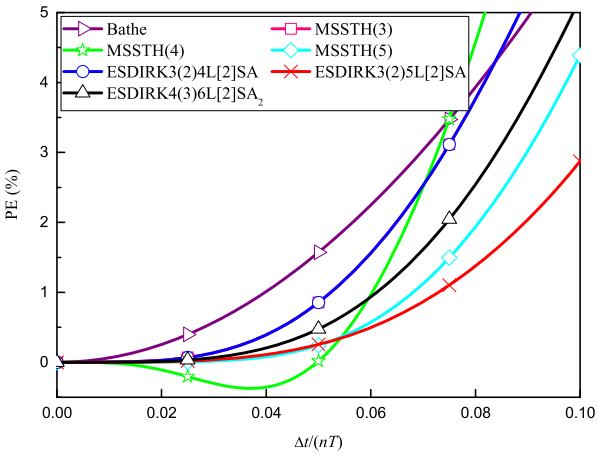

Fig. 6 Percentage period elongation of the higher-order methods with $\rho_\infty = 0.0$

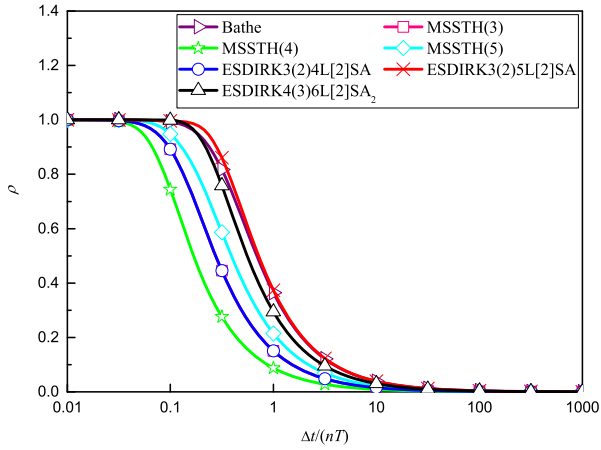

Fig. 7 Spectral radius of the higher-order methods with $\rho_\infty = 0.0$

5 Numerical experiments

The performance of the previously discussed integrators is illustrated in this Section by solving some benchmark problems in MBDyn.[3] Two values of ρ_∞, 0.0 and 0.6, are used for second-order methods, and $\rho_\infty = 0.0$ is used in MSSTH(3, 4, 5). In all examples, the integrators use the same $\Delta t/n$ to predict the results under comparable computational cost. The results of the second-order integrators with $\rho_\infty = 0.0$, the second-order integrators with $\rho_\infty = 0.6$, and the higher-order integrators, are presented separately. Table 1 lists the line and symbol of different integrators used in the result figures in this Section. The accuracy and stability of the numerical results as well as the calculation efficiency of the methods, are discussed.

[3]The input files for MBDyn are publicly available in the examples folder of the software distribution, https://public.gitlab.polimi.it/DAER/mbdyn.

Table 1 Line and symbol of different methods used in the figures in Sect. 5

Method	Line	Symbol	Method	Line	Symbol
Reference		×	LMS2		△
LMS3		◯	LMS4		◁
SS2		+	SS3		☆
SS4		◇	Bathe		▷
MSSTC(3)		▽	MSSTC(4)		□
MSSTC(5)		×			
Reference		×	MSSTH(3)		△
MSSTH(4)		◯	MSSTH(5)		◁
ESDIRK3(2)4L[2]S		+	ESDIRK3(3)5L[2]SA		☆
ESDIRK4(3)6L[2]SA$_2$		◇			

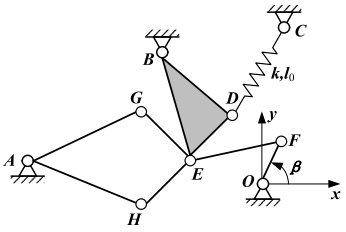

Fig. 8 Andrew's squeezer mechanism (adapted from [29])

5.1 Andrew's squeezer mechanism

Problem description Andrew's squeezer mechanism [29], as shown in Fig. 8, is a planar system composed of seven rigid links. The coordinates of noteworthy points in the local reference frame, the mass, and the moment of inertia of each part are listed in Table 2. The origin of each local reference frame is placed in the first point with the name of each link. For links with only two points, the local x-axis is along the line connecting the points. For the link E-B-D, the local y-axis is aligned with E-B, pointing towards B. The coordinates of the center of mass (x_C, y_C) are described in the local reference frame, and the rotational inertia I_z of each body is expressed about its centre of mass. In the global reference frame Oxy, the coordinates of points A, B, C are $(-0.06934\ \text{m}, -0.00227\ \text{m})$, $(-0.03635\ \text{m}, 0.03273\ \text{m})$ and $(0.01400\ \text{m}, 0.07200\ \text{m})$, respectively. The point D is connected to the point C by a spring, whose stiffness characteristic is $k = 4530$ N/m and whose natural length is $l_0 = 0.07785$ m. The link O-F is driven with a constant torque $T = 0.033\ \text{N} \cdot \text{m}$, starting from an initial angle $\beta_0 = -0.0620$ rad. Gravity is not considered.

Numerical results This example is used to check how the methods employed can preserve the mechanical energy of the system. The total energy balance equation of the system can be expressed as

$$\Delta E = E - E_0 - T(\beta - \beta_0) \tag{23}$$

since the torque T is constant, where E collects the kinetic and potential energy of the system. Without physical damping, ΔE should be zero throughout the simulation. With $\Delta t/n = 10^{-4}$ s, Figs. 9 and 10 show, respectively, the angular velocity ω of bar O-F, and

Table 2 Coordinates of the points in the local reference system, mass and inertia properties of the links

Link	Point	x (m)	y (m)	x_C (m)	y_C (m)	Mass (kg)	I_z (kg · m^2)
O-F	*O*	0.0	0.0	0.00092	0.0	0.04325	2.194×10^{-6}
	F	0.007	0.0				
E-F	*E*	0.0	0.0	0.01650	0.0	0.00365	4.410×10^{-7}
	F	0.028	0.0				
H-E	*H*	0.0	0.0	0.00579	0.0	0.00706	5.667×10^{-7}
	E	0.02	0.0				
G-E	*G*	0.0	0.0	0.00579	0.0	0.00706	5.667×10^{-7}
	E	0.02	0.0				
A-G	*A*	0.0	0.0	0.02308	0.00916	0.07050	1.169×10^{-5}
	G	0.04	0.0				
A-H	*A*	0.0	0.0	0.01228	−0.00449	0.05498	1.912×10^{-5}
	H	0.04	0.0				
E-B-D	*E*	0.0	0.0	0.01043	0.01626	0.02373	5.255×10^{-6}
	B	0.0	0.035				
	D	0.02	0.017				

ΔE as obtained from the numerical solutions within the $[0, 0.05]$ s interval. Figures 11 and 12 show the same results but with a smaller step size $\Delta t/n = 10^{-5}$ s. The results given by the second-order methods with $\rho_\infty = 0.0$ and 0.6, as well as the higher-order methods with $\rho_\infty = 0.0$, are presented separately. The reference solutions in Figs. 9 and 11 are obtained by LMS4 with $\rho_\infty = 0.6$ and $\Delta t = 10^{-6}$ s.

From Fig. 9, with $\Delta t/n = 10^{-4}$ s, the results computed using MSSTH(5) with $\rho_\infty = 0.0$ depart from the reference solution after about 0.01 s, and those obtained using MSSTH$(3, 4)$ with $\rho_\infty = 0.0$, ESDIRK3(2)4L[2]SA, and Bathe with $\rho_\infty = 0.6$ show observable differences. From Fig. 10, in this case, the results of ΔE given by MSSTC(5) with $\rho_\infty = 0.0$, MSSTH$(3, 4)$ with $\rho_\infty = 0.0$, and ESDIRK3(2)4L[2]SA, show an increasing trend, indicating that their results are likely to become unstable over time. LMS2 and SS2 with $\rho_\infty = 0.6$, LMS3, SS3, LMS4, SS4, MSSTC(3) and MSSTC(5) with $\rho_\infty = 0.6$, can mostly preserve energy, although with oscillations, while LMS2 and SS2 with $\rho_\infty = 0.0$, Bathe, MSSTC(4) and the remaining higher-order integrators show an obvious drop in energy.

With $\Delta t/n = 10^{-5}$ s, Fig. 11 shows that all methods can predict accurate results of ω. However, from Fig. 12, it is clear that the energy instability, i.e. $\Delta E > 0$ [5], is still observed for some higher-order integrators, including MSSTH$(3, 4)$ with $\rho_\infty = 0.0$, ESDIRK3(2)4L[2]SA, ESDIRK4(3)6L[2]SA$_2$. MSSTH(5) with $\rho_\infty = 0.0$ exhibits obvious energy drop. The second-order integrators perform well with the smaller step size.

From this example we can conclude that most of the higher-order integrators employed have worse stability in this type of problem, and are really not recommended for energy-conserving purpose. Among the second-order integrators, the linear analysis in Sect. 4 illustrates that they can preserve more modes as the number of steps or stages used increases, but the multi-stage methods do not follow this rule here. This is because they use the trapezoidal rule from the second to the nth stage, and the non-dissipative trapezoidal rule is likely to give unreliable or unstable results for nonlinear problems [33]. Therefore, the multi-step

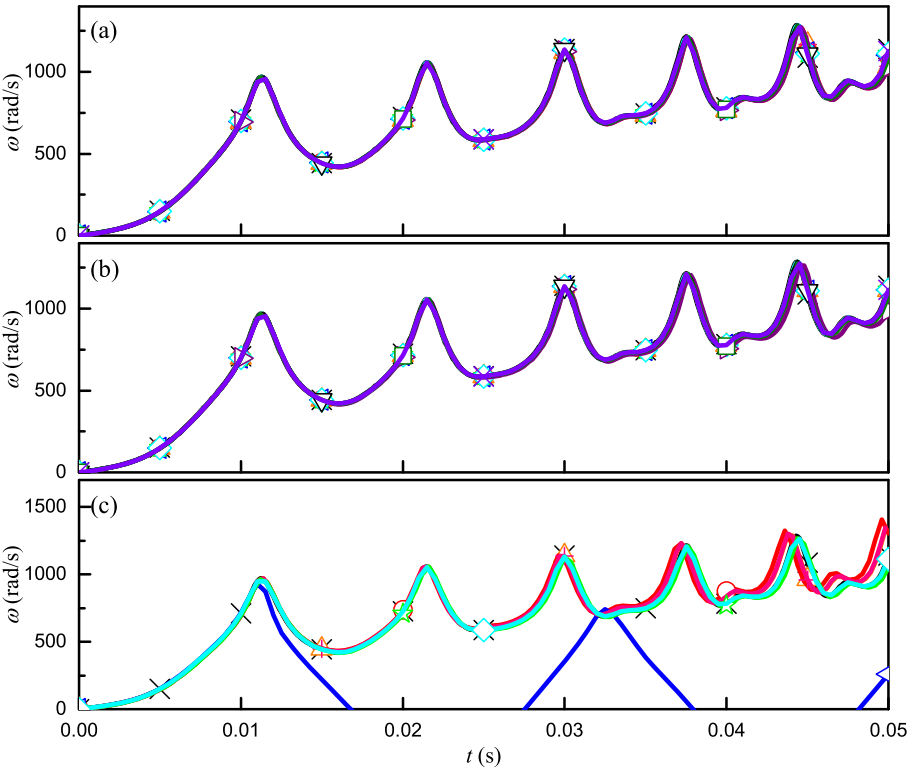

Fig. 9 Angular velocity ω of bar O-F within $[0, 0.05]$ s using $\Delta t/n = 10^{-4}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

and single-step integrators, especially LMS3, SS3, LMS4 and SS4 with a large ρ_∞, such as 0.6, are better candidates in terms of energy conservation for general problems.

5.2 Flexible four-bar mechanism

Problem description Some of the benchmark problems for flexible mechanisms proposed in [4] are solved. Figure 13 shows the configuration of a flexible four-bar mechanism. Bars A-B, B-C, C-D and the ground are connected through revolute joints. The initial angles between them are all 90 deg. The bars' lengths are $L_1 = L_3 = 0.12$ m and $L_2 = 0.24$ m. The rotation axis of the revolute joint at point C is rotated by $+5$ deg about the y direction, to simulate an assembly defect that would lock the mechanism if the bars were rigid. The inertia and stiffness properties of the bars are listed in Table 3. Each bar is modeled in MBDyn using 5 three-node beam elements [15]. The angular velocity of the bar A-B at point A with respect to the frame is prescribed as $\Omega = 0.6$ rad/s during the simulation.

Numerical results The simulation was run in the interval $[-2T, 12$ s$]$, where $T = 2\pi/\Omega$ performs about three complete cycles of the bar A-B. With $\Delta t/n = 4 \times 10^{-3}$ s, Figs. 14 and 15 respectively show the rotation angle θ and the angular velocity ω_1 about the x axis at the tip of bar B-C at point C within $[0, 12]$ s. Figure 16 plots the results of ω_1 with a smaller step size $\Delta t = 1 \times 10^{-3}$ s.

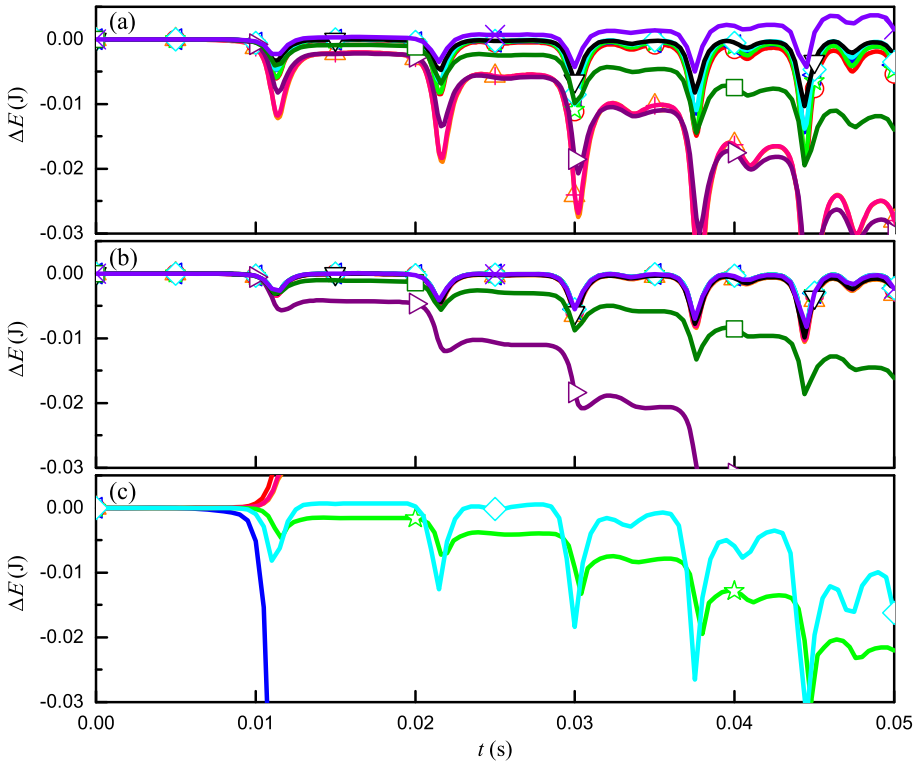

Fig. 10 Energy balance ΔE within $[0, 0.05]$ s using $\Delta t/n = 10^{-4}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

Table 3 Inertia and stiffness properties of bars A-B, B-C and C-D

	Bar A-B	Bar B-C	Bar C-D
Mass per unit span m (kg/m)	1.9968	0.4992	1.9968
Moments of inertia per unit span J_1 (mg · m)	85.1968	5.3248	85.1968
Moments of inertia per unit span J_2 (mg · m)	42.5984	2.6624	42.5984
Moments of inertia per unit span J_3 (mg · m)	42.5984	2.6624	42.5984
Axial stiffness EA (MN)	52.9920	13.2480	52.9920
Shearing stiffness GA_Y (MN)	16.8803	4.2201	16.8803
Shearing stiffness GA_Z (MN)	16.8803	4.2201	16.8803
Torsional stiffness GJ (N · m^2)	733.488	45.843	733.488
Bending stiffness EJ_Y (N · m^2)	1130.50	70.656	1130.50
Bending stiffness EJ_Z (N · m^2)	1130.50	70.656	1130.50

From Fig. 14 it is seen that the rotation angles θ computed by the methods used agree very well with each other. However, as shown in Figs. 15 and 16, since the angular velocity changes rapidly at certain moments, high-frequency oscillations can be observed, and the

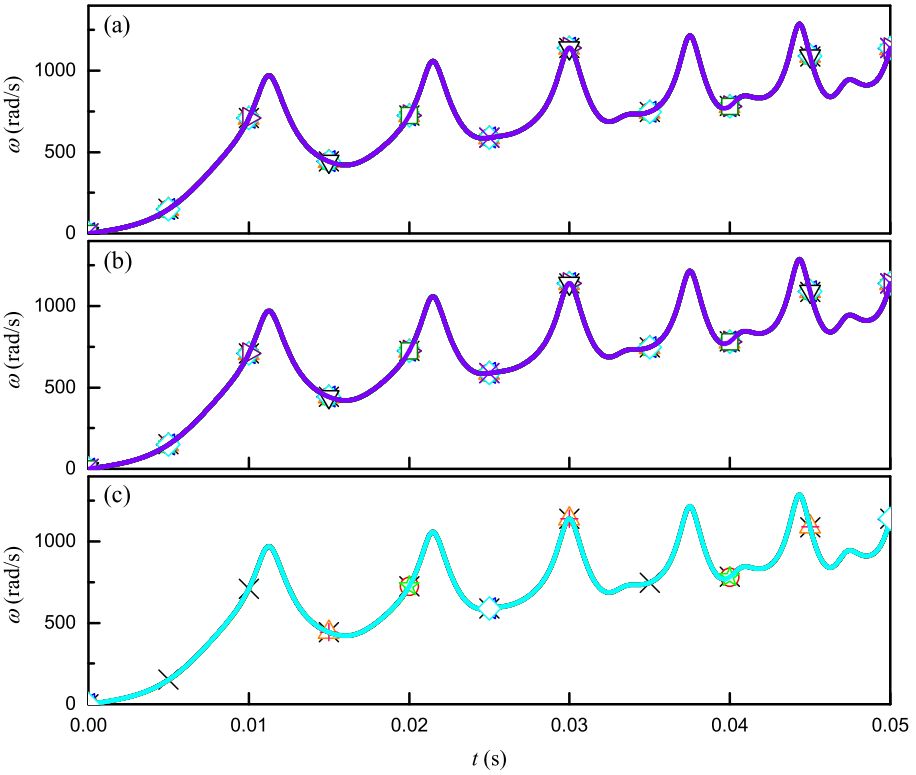

Fig. 11 Angular velocity ω of bar O-F within $[0, 0.05]$ s using $\Delta t/n = 10^{-5}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

observable differences in the integrators employed are how fast they filter the oscillations. As shown in Figs. 15 and 16, the oscillations become more pronounced with a smaller time step size or a larger ρ_∞. LMS4 and SS4 with $\rho_\infty = 0.6$ and $\Delta t = 1 \times 10^{-3}$ s exhibit the most significant oscillations. This example shows the importance of algorithmic dissipation for problems containing high-frequency pollution, which often appears in the solutions of velocities, accelerations and forces. For such problems, Bathe and the higher-order integrators, which have strong algorithmic dissipation from linear analysis in Sect. 4, with $\rho_\infty = 0.0$ and a suitable Δt, not too small, are recommended.

5.3 Lateral buckling of a thin beam

Problem description Figure 17 shows the configuration of a beam actuated by a crank-link mechanism. The beam is clamped at one end; the other end is connected to the link by a spherical joint. To simulate an initial imperfection, the plane of the crank-link mechanism is offset from the plane of the beam by $d = 0.1$ mm in the y direction. The end of the beam and the spherical joint are rigidly connected. The link, crank, and ground are connected via revolute joints. The lengths are $L = 1$ m, $L_l = 0.25$ m and $L_c = 0.05$ m. The inertia and stiffness properties of the crank, link and beam are listed in Table 4. The beam is meshed with 5 three-node beam elements, and both the link and the crank are modelled with 1 three-

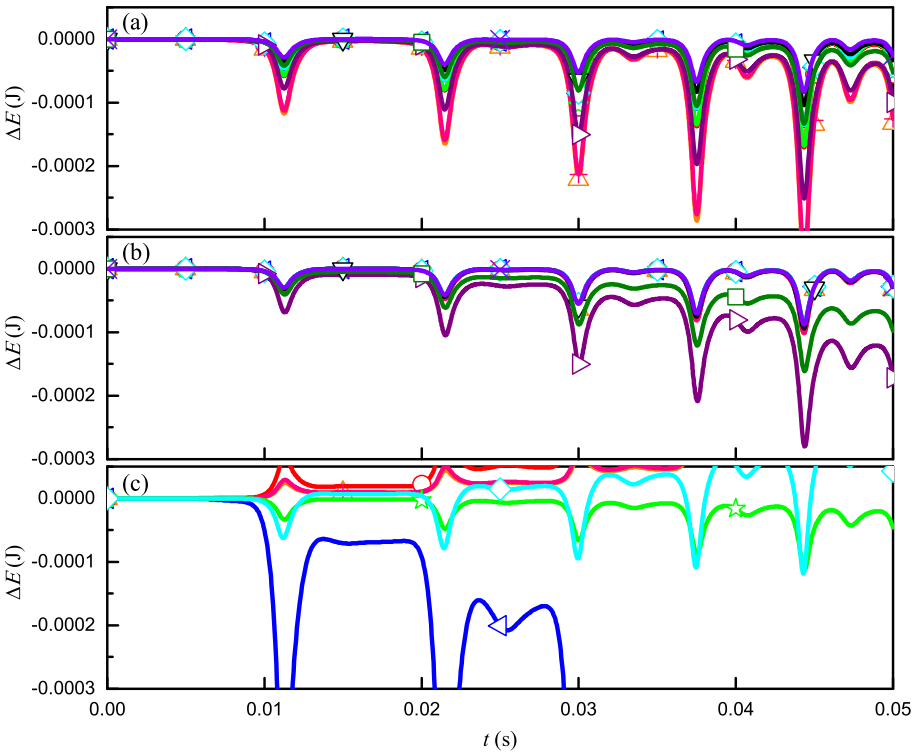

Fig. 12 Energy balance ΔE within $[0, 0.05]$ s using $\Delta t/n = 10^{-5}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

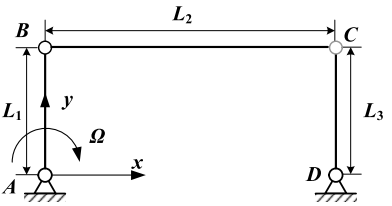

Fig. 13 Flexible four-bar mechanism (adapted from [4])

node beam element each. The rotation angle of the crank is prescribed as

$$\phi = \begin{cases} \dfrac{\pi}{2}\left(1 - \cos\dfrac{\pi t}{T}\right), & t \le T \\ \pi, & t > T \end{cases} \tag{24}$$

where $T = 0.4$ s.

Numerical results The simulation was run in the interval $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s. The rotation angle θ and angular velocity ω_1 about the x axis, and the shear force F_3 along the z axis at the mid-span of the beam are summarized in Figs. 18, 19 and 20, respectively. Driven by the crank-link mechanism, the beam buckled laterally rather quickly, before the

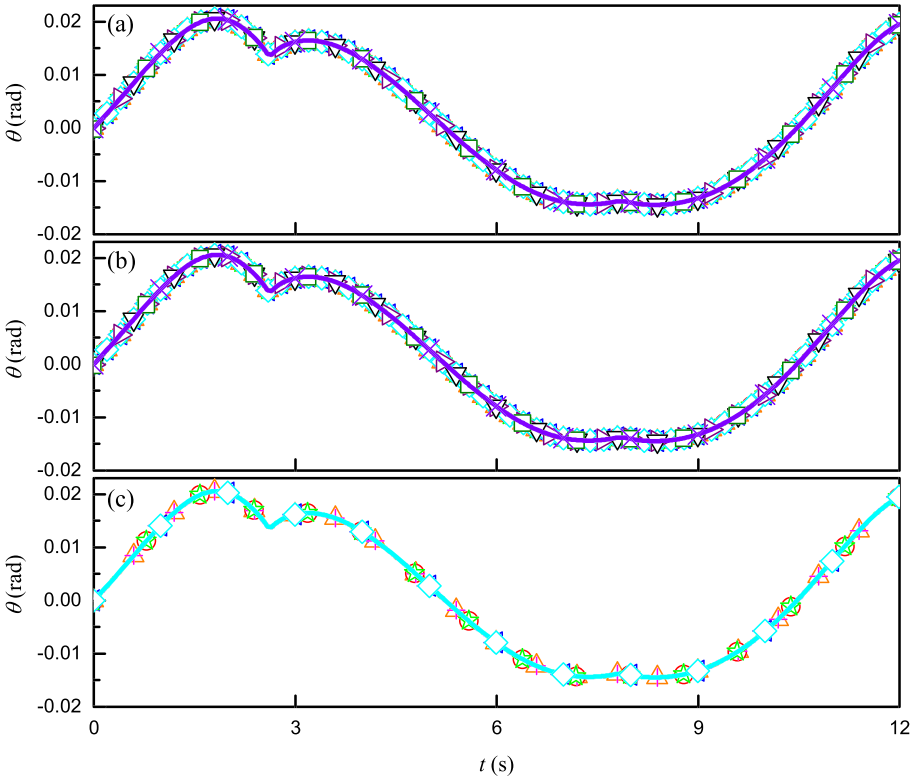

Fig. 14 Rotation angle θ about the x axis at the tip of bar B-C at point C within $[0, 12]$ s using $\Delta t/n = 4 \times 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

crank reached the uppermost position, and then began to oscillate rapidly. In this case, the higher-order MSSTH(5) with $\rho_\infty = 0.0$, and ESDIRK3(3)5L[2]SA failed to complete the simulation.

As shown in Figs. 18–20, the solutions of the employed methods almost overlap in the $[0, 0.4]$ s interval. However, in the subsequent free oscillations, the solutions obtained with a few methods, including LMS2 and SS2 with $\rho_\infty = 0.0$, MSSTH(3, 4), ESDIRK3(2)4L[2]SA, exhibit a significant amplitude decay and phase shift compared to the solutions obtained with the remaining methods. These schemes cannot reflect the participation of high-frequency contributions in the response, because of their strong algorithmic dissipation. The results are consistent with the comparisons in Sect. 4. LMS2 and SS2 with $\rho_\infty = 0.0$ and most of the higher-order schemes have larger algorithmic dissipation than the other schemes. Therefore, when the contribution of high-frequencies needs to be considered in the solution, the second-order methods, especially LMS3, SS3, LMS4, SS4 with a large ρ_∞, such as 0.6, are recommended.

5.4 Rotating shaft

Problem description Figure 21 shows the configuration of a rotating shaft. Its end R is connected to the ground by a revolute joint, whereas the other end T is connected to the

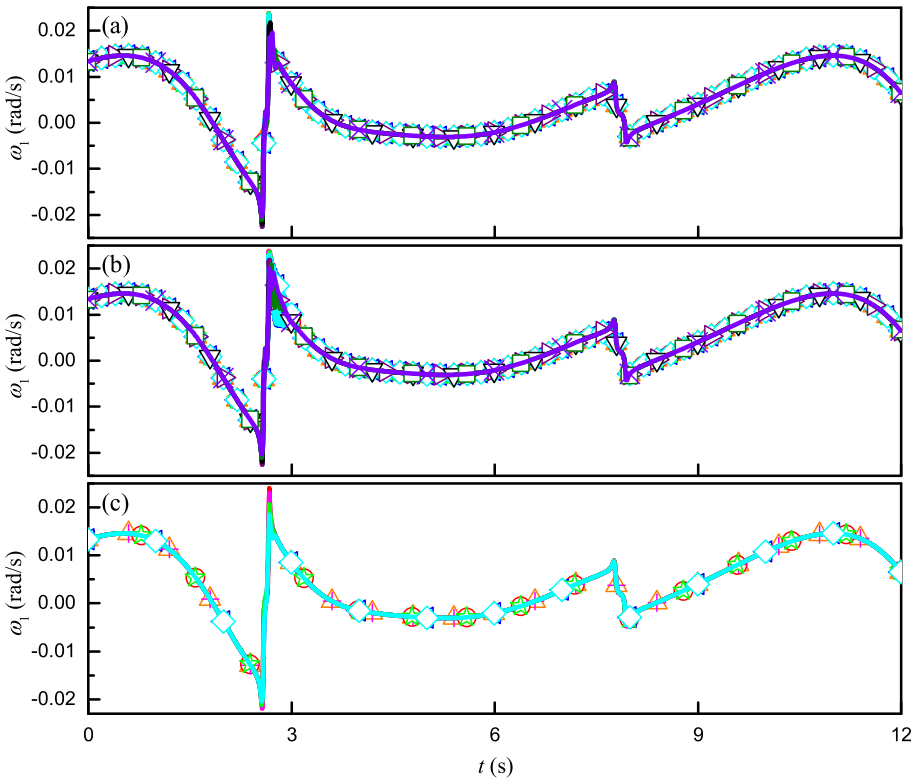

Fig. 15 Angular velocity ω_1 about the x axis at the tip of bar B-C at point C within $[0, 12]$ s using $\Delta t/n = 4 \times 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

ground by a cylindrical joint that allows rotation about and displacement along the shaft's axis. A rigid disk is attached to the shaft at the mid-span; its center of mass is offset from the reference axis of the shaft by $d = 0.05$ m in the z direction, thus inertially unbalancing the system. The shaft length is $L = 6$ m; its other properties are listed in Table 5. It is modelled with 16 three-node beam elements. The disk has mass $m_d = 70.573$ kg, radius $r_d = 0.24$ m, and thickness $t_d = 0.05$ m. Its inertial tensor with respect to the center of mass is diag$(2.0325, 1.0163, 1.0163)$ g $\cdot$ m^2. The angular velocity of the revolute joint at the point R is prescribed as

$$\Omega = \begin{cases} \dfrac{1}{2}A_1\omega\left(1 - \cos\dfrac{\pi t}{T_1}\right), & 0 \le t \le T_1 \\ A_1\omega, & T_1 < t \le T_2 \\ A_1\omega + \dfrac{1}{2}(A_2 - A_1)\omega\left(1 - \cos\dfrac{\pi(t - T_2)}{T_3 - T_2}\right), & T_2 < t \le T_3 \\ A_2\omega, & t > T_3 \end{cases} \tag{25}$$

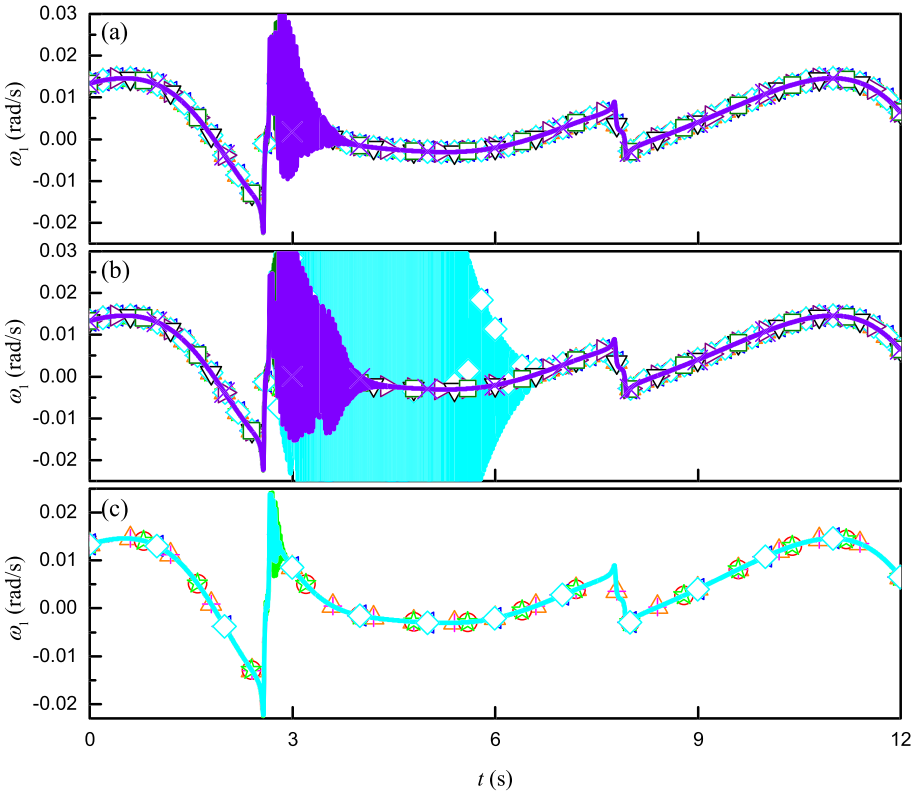

Fig. 16 Angular velocity ω_1 about the x axis at the tip of bar B-C at point C within [0, 12] s using $\Delta t/n = 1 \times 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

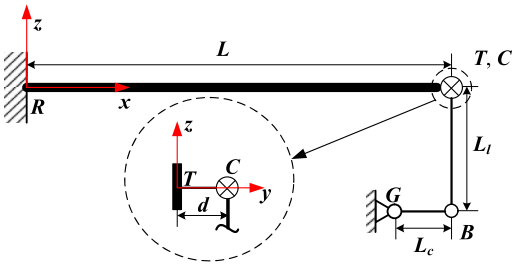

Fig. 17 Crank actuated beam (adapted from [4])

where $A_1 = 0.8$, $A_2 = 1.2$, $T_1 = 0.5$ s, $T_2 = 1$ s, $T_3 = 1.25$ s, $\omega = 60$ rad/s, i.e., it gently grows to an amplitude slightly below ω, pauses there for some time, and then gently grows again above ω, which corresponds roughly to a characteristic frequency of the system.

Numerical results The simulation was run in the interval $[-1, 2.5]$ s using $\Delta t/n = 10^{-3}$ s. Figures 22, 23 and 24 show, respectively, the displacement u_3, velocity v_3 and acceleration a_3 along the z axis at the shaft's mid-span within [0, 2.5] s. In this case, LMS4 with $\rho_\infty = 0.6$ failed to complete the solution. Excellent agreement can be observed between the results

Table 4 Inertia and stiffness properties of the crank, link and beam

	Beam	Link	Crank
Mass per unit span m (kg/m)	2.68	1.212	4.85
Moments of inertia per unit span J_1 (mg · m)	2255.33	87.30	1396.6
Moments of inertia per unit span J_2 (mg · m)	2233	43.65	698.3
Moments of inertia per unit span J_3 (mg · m)	22.33	43.65	698.3
Axial stiffness EA (MN)	73	33.02	132.1
Shearing stiffness GA_Y (MN)	5.025	10.81	43.22
Shearing stiffness GA_Z (MN)	23.40	10.81	43.22
Torsional stiffness GJ (N · m^2)	877.2	914.5	14630
Bending stiffness EJ_Y (N · m^2)	60830	1189	19020
Bending stiffness EJ_Z (N · m^2)	608.3	1189	19020

Table 5 Inertia and stiffness properties of the shaft

	Shaft
Mass per unit span m (kg/m)	11.64
Moments of inertia per unit span J_1 (mg · m)	26.34
Moments of inertia per unit span J_2 (mg · m)	13.17
Moments of inertia per unit span J_3 (mg · m)	13.17
Axial stiffness EA (MN)	313.4
Shearing stiffness GA_Y (MN)	60.5
Shearing stiffness GA_Z (MN)	60.5
Torsional stiffness GJ (N · m^2)	272.7
Bending stiffness EJ_Y (N · m^2)	354.5
Bending stiffness EJ_Z (N · m^2)	354.5

computed by the other employed methods, since the time step is sufficient to accurately describe the participating modes regardless of the algorithmic dissipation.

Convergence Based on this example, the convergence rates of the employed methods are investigated. Figures 25, 26 and 27 plot the relative errors of u_3, v_3 and a_3 versus $\Delta t/n$ of the employed methods. The relative error RE is defined as

$$\mathrm{RE}(x) = \sqrt{\frac{\sum_{j=1}^{N}(x(t_j) - x_j)^2}{\sum_{j=1}^{N} x(t_j)^2}} \tag{26}$$

where N is the number of total steps, $x(t_j)$ and x_j denote the exact and numerical solutions at time t_j, respectively. Since no exact solution can be obtained analytically for problems like this, it is replaced by the reference solution obtained using ESDIRK4(3)6L[2]SA$_2$ with a very small step size, $\Delta t = 10^{-5}$ s.

As can be seen, the second-order methods all exhibit a second-order convergence rate for displacement, velocity, and acceleration. The multi-step and equivalent single-step schemes

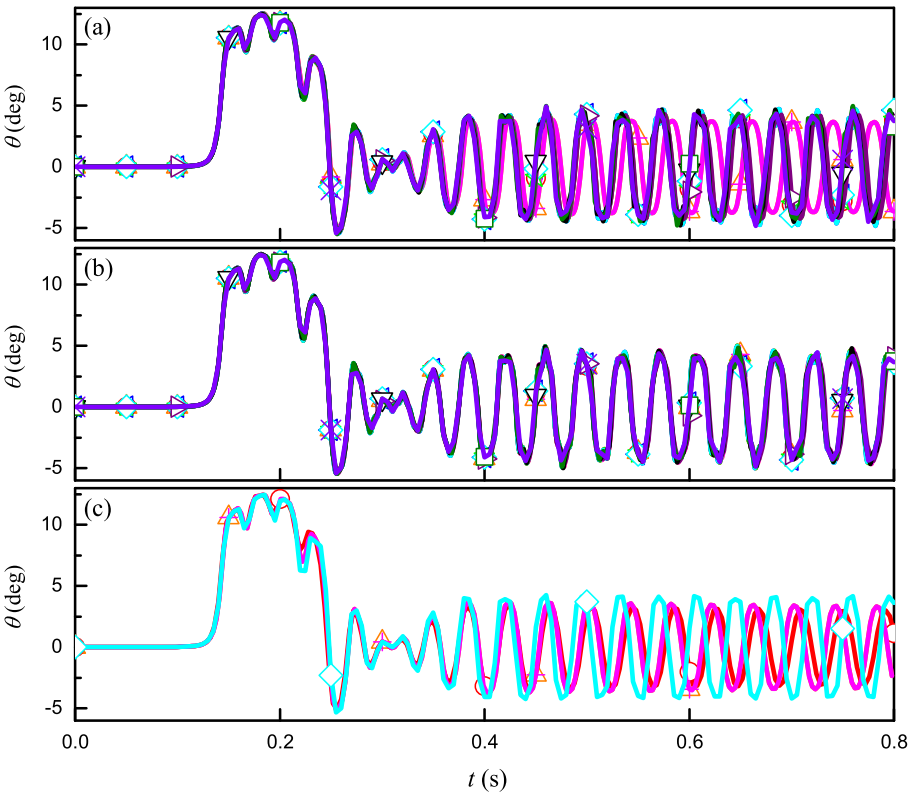

Fig. 18 Rotation angle θ about the x axis at the beam's mid-span within $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

have very similar relative errors. Accuracy improves with increasing number of steps or stages or with a large ρ_∞. LMS2 and SS2 with $\rho_\infty = 0.0$ show larger errors than the other schemes. These conclusions are consistent with the analysis in Sect. 4.

Among the higher-order methods, the third-order MSSTH(3) and ESDIRK3(2)4L[2]SA show a third-order convergence rate, and the third-order ESDIRK3(3)5L[2]SA shows a convergence rate exceeding the third-order. The fourth-order MSSTH(4) and ESDIRK4(3)6L[2]SA$_2$ show a convergence rate of about fourth-order when $\Delta t/n$ is close to 0.001 s, but when $\Delta t/n$ becomes smaller, their orders of convergence rate decrease, especially in a_3. We think that this is because the reference solution, obtained using a very small step size, includes the contribution of a very large frequency range that contains many high frequencies not included in the numerical results. The treatment of these high-frequencies brings additional errors, which become more significant in the accelerations and in the cases where the error of the integrator in the frequency domain that it can retain is already quite small, such as the higher-order integrators with a small step size.

However, the fifth-order MSSTH(5) was never able to reach the fifth-order convergence rate. Regarding the reason, we think it may be related to the smoothness of the problem itself. Since the accuracy order is obtained from the Taylor expansion results, the higher-order

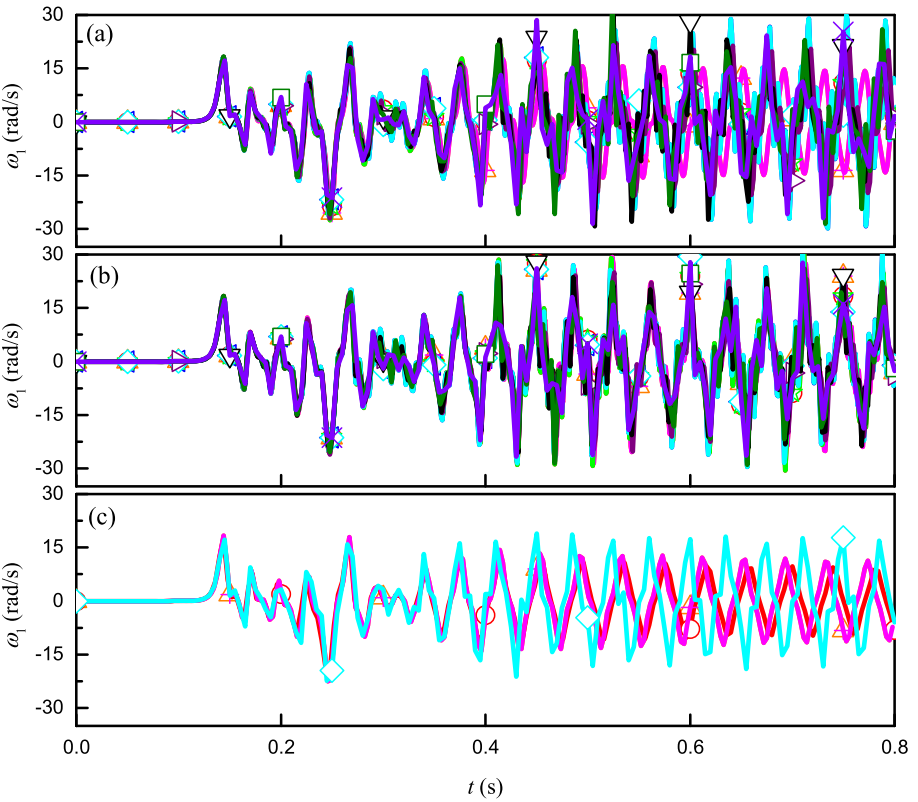

Fig. 19 Angular velocity ω_1 about the x axis at the beam's mid-span within $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

methods have higher requirements on the smoothness of the problems. If the smoothness does not meet the requirements, the order reduction may occur.

From the relative errors, ESDIRK3(3)5L[2]SA and ESDIRK4(3)6L[2]SA$_2$ are more accurate than other methods. The design of MSSTH(3, 4, 5) considers the minimization of the local truncation error, but these methods do not perform as expected. It indicates that a small local truncation error does not necessarily mean higher accuracy. The amplitude and period accuracy, as shown in Figs. 5–6, can better represent the accuracy of an integrator. Therefore, for high-accuracy purpose, ESDIRK3(3)5L[2]SA and ESDIRK4(3)6L[2]SA$_2$ are more recommended.

5.5 Average number of iterations

Since the same $\Delta t/n$ is used in the examples, the number of steps or sub-steps required for all methods is the same. However, as discussed in Sect. 2, the scheme used for prediction also plays an important role in computational efficiency. Therefore, the average number of iterations required for each step/sub-step in all examples solved in this section is listed in Fig. 28, to illustrate the cost of the employed methods. Two time step sizes are considered here for each example.

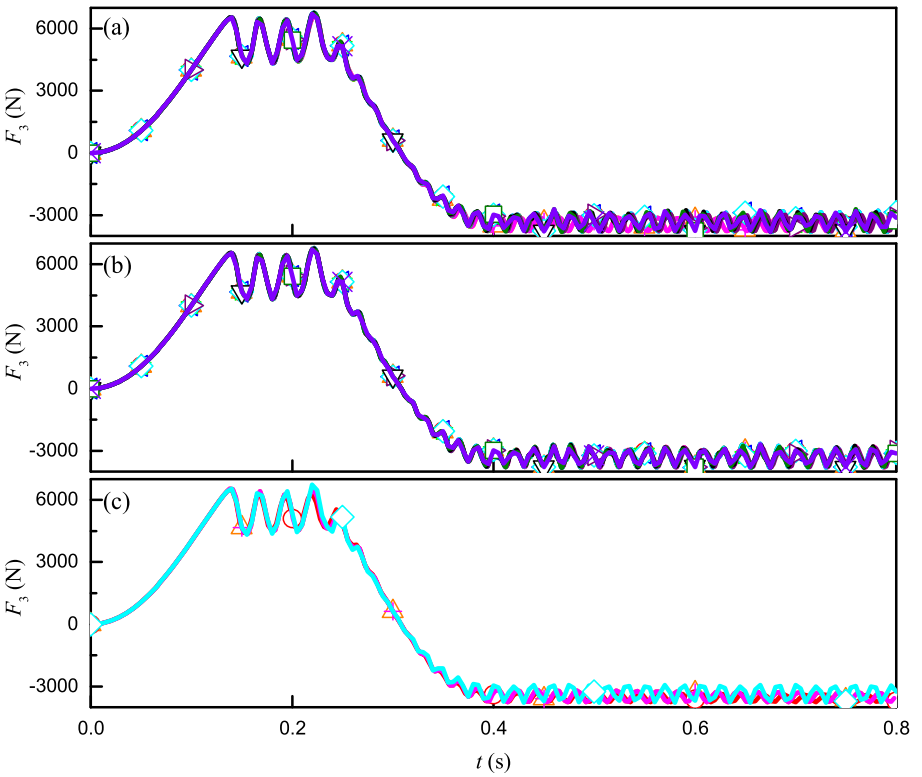

Fig. 20 Shear force F_3 at the beam's mid-span within $[0, 0.8]$ s using $\Delta t/n = 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

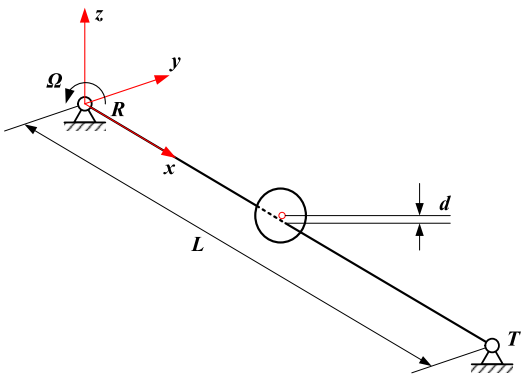

Fig. 21 Rotating shaft with unbalanced disk (adapted from [4])

As shown in Fig. 28, the average number of iterations decreases with a smaller step size in all cases. The number of iterations spent by each integrator does not differ significantly. However, it can be observed that each single-step method always requires more iterations than the corresponding multi-step method. As discussed in Sect. 3, single-step methods use constant prediction, while multi-step methods use an explicit second-order scheme for

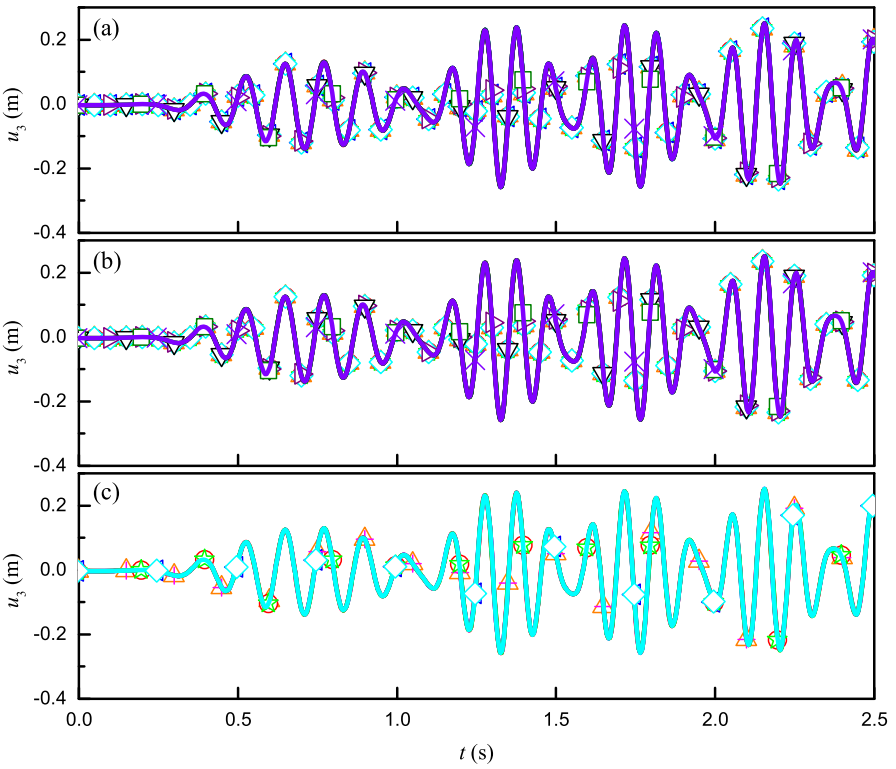

Fig. 22 Displacement u_3 at the shaft's mid-span within $[0, 2.5]$ s using $\Delta t/n = 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

prediction. Therefore, this observation supports the conclusion that the use of an explicit second-order prediction scheme in these second-order methods helps to improve computational efficiency.

Besides, also the higher-order methods require more iterations than the remaining second-order methods in most cases. This may also be explained by the consideration that the accuracy of the prediction scheme is not close enough to that of the time integration scheme. The second-order multi-step and multi-stage methods usually require the least number of iterations of all methods. The data from Fig. 28 seem to support the consideration that a suitable prediction scheme, which must be explicit and have accuracy close to that of the integration scheme, is helpful in saving computational costs.

6 Conclusions

In this work, the performance of several representative implicit, A-stable time integration methods is discussed in view of their application to multibody system dynamics. The employed methods include linear two-, three-, and four-step methods, referred to as LMS2, LMS3, LMS4, their equivalent single-step methods, indicated as SS2, SS3, SS4, and several explicit first-stage, singly diagonally-implicit Runge–Kutta methods (ESDIRKs), indicated as Bathe, MSSTC(3, 4, 5), MSSTH(3, 4, 5), ESDIRK3(2)4L[2]SA, ESDIRK3(3)5L[2]SA,

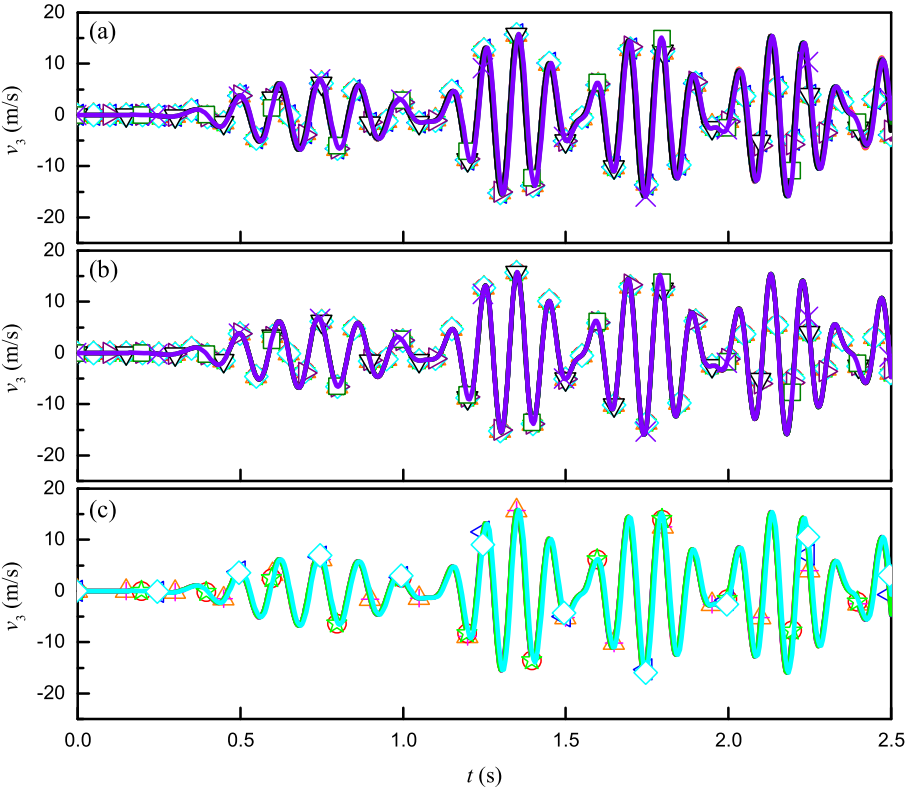

Fig. 23 Velocity v_3 at the shaft's mid-span within $[0, 2.5]$ s using $\Delta t/n = 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

and ESDIRK4(3)6L[2]SA$_2$. These methods have been developed in [21, 35, 36], but the parameters of MSSTH(3, 4, 5) are modified here to satisfy the overall and stage order conditions. The formulations of the employed methods, and their implementation in the free general-purpose multibody solver MBDyn are presented.

In terms of properties, the linear multi-step, single-step, Bathe and MSSTC(3, 4, 5) methods have second-order accuracy and tunable algorithmic dissipation, whereas the other ESDIRKs can achieve higher-order accuracy. Several general conclusions can be drawn from the linear analysis and numerical experiments:

- with a suitable step size, all employed methods can predict accurate solutions.
- LMS3, SS3, LMS4 and SS4 with a large ρ_∞, such as 0.6, show robust stability and good energy-conserving properties, making them suitable for long-term simulations and other cases where a large range of modes must be preserved, but these methods are not as good as others at filtering out high-frequency oscillations.
- LMS2 and SS2 with $\rho_\infty = 0.0$ (namely the second-order Backward Difference Formula) as well as most of the employed higher-order methods show a strong algorithmic dissipation even in the low-frequency range, so that their solutions are more likely to exhibit obvious amplitude decay and consequently a loss of accuracy at the large time steps.
- Bathe and the higher-order integrators with $\rho_\infty = 0.0$ can filter out high-frequency dynamics faster, so they are more useful for problems with high-frequency pollution.

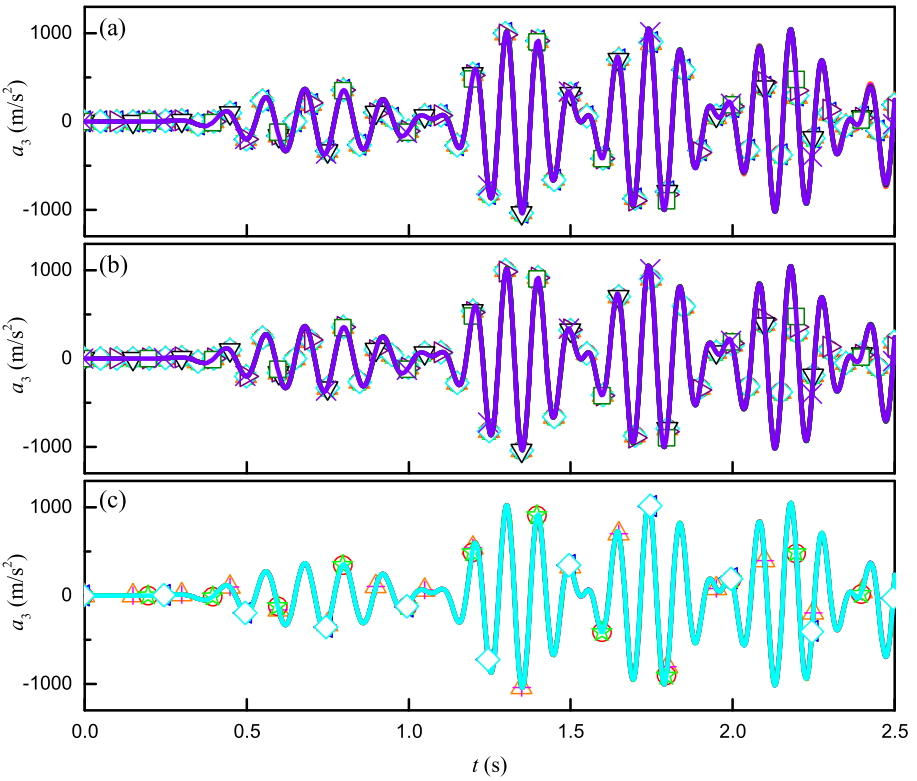

Fig. 24 Acceleration a_3 at the shaft's mid-span within $[0, 2.5]$ s using $\Delta t/n = 10^{-3}$ s (**a**) Second-order methods with $\rho_\infty = 0.0$; (**b**) Second-order methods with $\rho_\infty = 0.6$; (**c**) Higher-order methods with $\rho_\infty = 0.0$

– among the employed methods, ESDIRK3(3)5L[2]SA and ESDIRK4(3)6L[2]SA$_2$ have an obvious accuracy advantage over the other when the time step size is small enough to correctly integrate the participating dynamics, so they are recommended for high-accuracy purposes.
– the prediction scheme affects the number of iterations and thus the computational efficiency. It should be explicit, preferably if its order of accuracy is close to that of the time integration scheme.

Appendix: Modified parameters of MSSTH(3, 4, 5)

Following the approach in [21], the local truncation error of an ESDIRK can be expressed as

$$\sigma = \sum_{i=1}^{\infty} (\Delta t)^i \sum_{j=1}^{a_i} \sigma_j^{(i)} F_j^{(i)} \tag{27}$$

where $F_j^{(i)}$ are elementary differentials, $a_i = \{1, 1, 2, 3, 9, 20\}$ for $i = \{1, 2, 3, 4, 5, 6\}$. The method is said to be pth-order accurate if $\sigma = O(\Delta t^{p+1})$, which requires that $\sigma_j^{(i)} = 0$ for

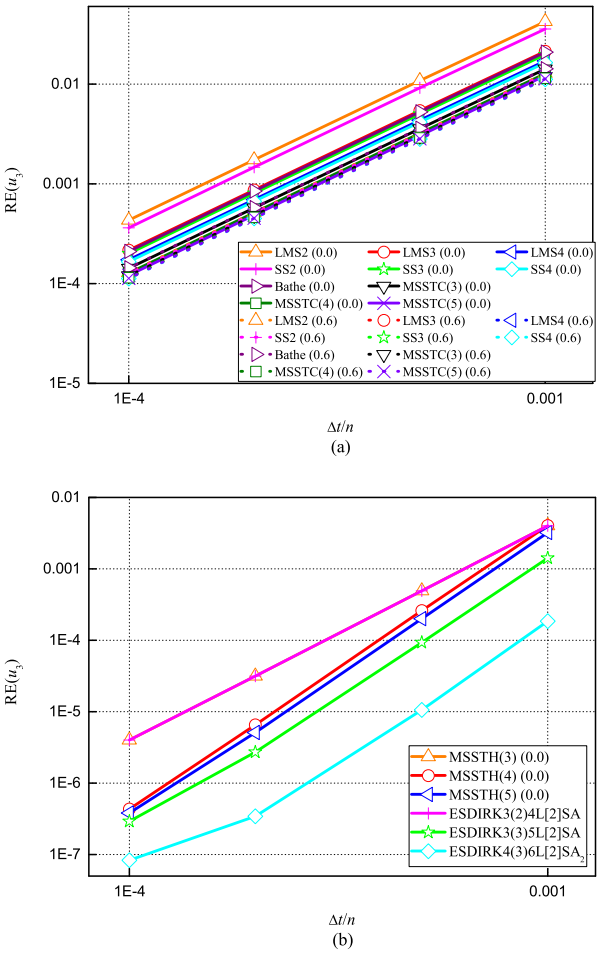

Fig. 25 Convergence rates of displacement u_3 at the shaft's mid-span for: (**a**) second-order methods; (**b**) higher-order methods (the value of ρ_∞ is placed in brackets in the legend)

all $j = 1, 2, \ldots, a_i$ and $i = 1, 2, \ldots, p$. The error of a pth-order method can be measured by

$$||\boldsymbol{\sigma}^{(p+1)}||_2 = \sqrt{\sum_{j=1}^{a_{p+1}} (\sigma_j^{(p+1)})^2} \tag{28}$$

Here we use $\boldsymbol{A} = a_{ij}$, $\boldsymbol{b} = b_i$, $\boldsymbol{c} = c_i$, $\boldsymbol{C} = \text{diag}(\boldsymbol{c})$, $\boldsymbol{e} = [1, 1, 1, \ldots, 1]^{\text{T}}$, and define

$$\boldsymbol{q}^{(k)} = \boldsymbol{A}\boldsymbol{C}^{k-1}\boldsymbol{e} - \frac{1}{k}\boldsymbol{C}^k\boldsymbol{e},\ \boldsymbol{Q}^{(k)} = \text{diag}(\boldsymbol{q}^{(k)}) \tag{29}$$

the overall order conditions for up to sixth-order are expressed in Table 6. If $\boldsymbol{q}^{(1)} = \boldsymbol{q}^{(2)} = \cdots = \boldsymbol{q}^{(\ell)} = \boldsymbol{0}$, the stage order of an ESDIRk is ℓ. According to the definition, one can check that MSSTC(n) has an overall order 2 and stage order 2.

MSSTH(n) is designed to provide high accuracy, so the conditions of stage order 2, overall order n, and a $||\sigma^{(n+1)}||_2$ as small as possible are imposed. For MSSTH(3), $\boldsymbol{q}^{(1)} =$

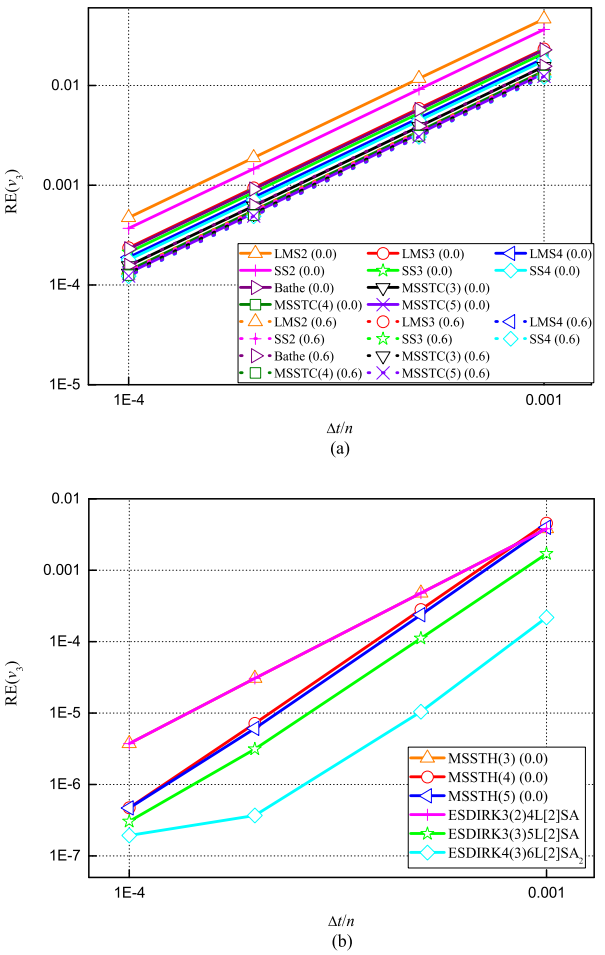

Fig. 26 Convergence rates of velocity v_3 at the shaft's mid-span for: (**a**) second-order methods; (**b**) higher-order methods (the value of ρ_∞ is placed in brackets in the legend)

$\boldsymbol{q}^{(2)} = \mathbf{0}$, and $||\boldsymbol{\sigma}^{(1)}||_2 = ||\boldsymbol{\sigma}^{(2)}||_2 = ||\boldsymbol{\sigma}^{(3)}||_2 = 0$ yield

$$c_2 = 2\gamma \tag{30a}$$

$$a_{21} = \gamma \tag{30b}$$

$$a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma} \tag{30c}$$

$$a_{31} = c_3 - \gamma - a_{32} \tag{30d}$$

$$b_2 = \frac{3c_3 + 6\gamma - 6c_3\gamma - 2}{12\gamma(c_3 - 2\gamma)} \tag{30e}$$

$$b_3 = \frac{6\gamma^2 - 6\gamma + 1}{3c_3(c_3 - 2\gamma)} \tag{30f}$$

$$b_1 = 1 - \gamma - b_2 - b_3 \tag{30g}$$

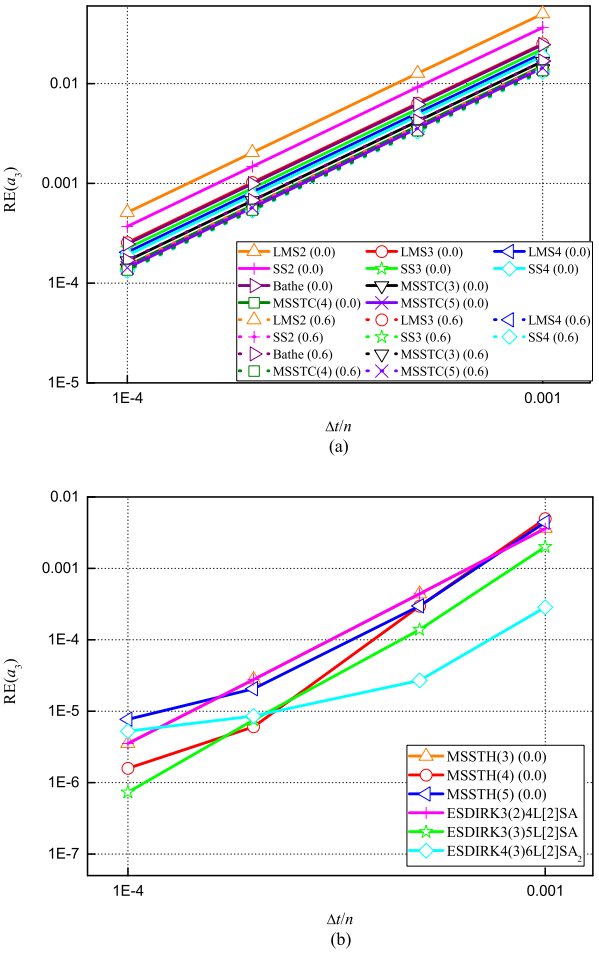

Fig. 27 Convergence rates of acceleration a_3 at the shaft's mid-span for: (**a**) second-order methods; (**b**) higher-order methods (the value of ρ_∞ is placed in brackets in the legend)

leaving two free parameters γ and c_3. For a given ρ_∞, γ has been determined in [35] to offer the prescribed degree of algorithmic dissipation, as listed in Table 7, and c_3 can be obtained by minimizing $||\boldsymbol{\sigma}^{(4)}||$ as

$$c_3 = \frac{24\gamma^2 - 20\gamma + 3}{24\gamma^2 - 24\gamma + 4} \tag{31}$$

For MSSTH(4), the order conditions $\boldsymbol{q}^{(1)} = \boldsymbol{q}^{(2)} = \boldsymbol{0}$, and $||\boldsymbol{\sigma}^{(1)}||_2 = ||\boldsymbol{\sigma}^{(2)}||_2 = ||\boldsymbol{\sigma}^{(3)}||_2 = ||\boldsymbol{\sigma}^{(4)}||_2 = 0$ give

$$c_2 = 2\gamma \tag{32a}$$

$$a_{21} = \gamma \tag{32b}$$

$$a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma} \tag{32c}$$

$$a_{31} = c_3 - \gamma - a_{32} \tag{32d}$$

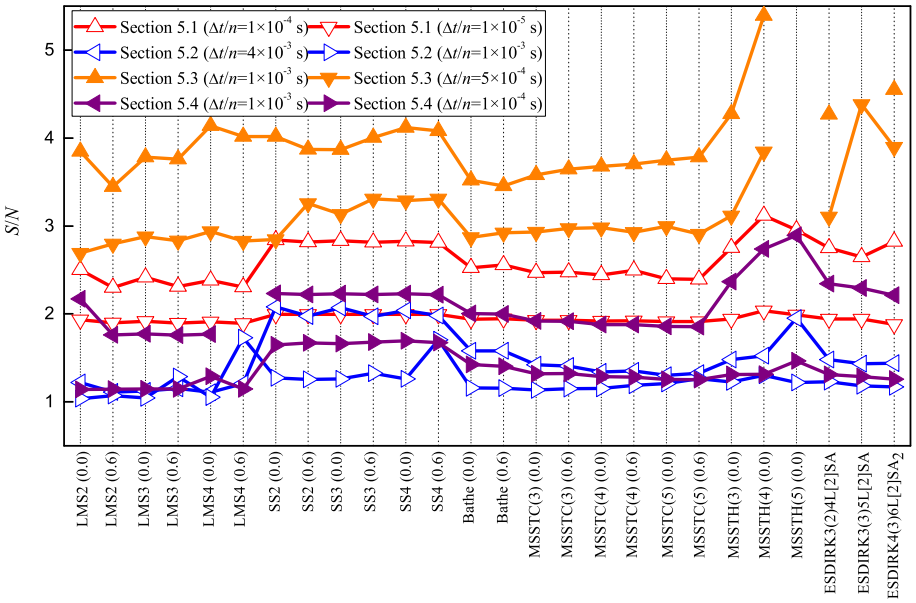

Fig. 28 Average number of iterations required in each step/sub-step S/N of the integrators in all examples (S is the total number of iterations, and N is the total number of steps/sub-steps)

$$a_{42} = \frac{c_4(c_4 - 2\gamma)\eta_1}{4\gamma\eta_2} \tag{32e}$$

$$a_{43} = \frac{c_4^2 - 4a_{42}\gamma - 2c_4\gamma}{2c_3} \tag{32f}$$

$$a_{41} = c_4 - \gamma - a_{42} - a_{43} \tag{32g}$$

$$b_2 = -\frac{12(c_3c_4 - c_3 - c_4 + 1)\gamma + 4c_3 + 4c_4 - 6c_3c_4 - 3}{24\gamma(c_3 - 2\gamma)(c_4 - 2\gamma)} \tag{32h}$$

$$b_3 = \frac{24(c_4 - 1)\gamma^2 + 4(5 - 6c_4)\gamma + 4c_4 - 3}{12c_3(c_4 - c_3)(c_3 - 2\gamma)} \tag{32i}$$

$$b_4 = -\frac{24(c_3 - 1)\gamma^2 + 4(5 - 6c_3)\gamma + 4c_3 - 3}{12c_4(c_4 - c_3)(c_4 - 2\gamma)} \tag{32j}$$

$$b_1 = 1 - \gamma - b_2 - b_3 - b_4 \tag{32k}$$

with

$$\begin{aligned}\eta_1 = {} & 48(1 - c_4)\gamma^3 + 8(3c_3^2 - 6c_3 + 9c_4 - 5)\gamma^2 \\ & + 6(-4c_3^2 + 6c_3 - 4c_4 + 1)\gamma + 4c_3^2 - 5c_3 + 2c_4\end{aligned} \tag{33a}$$

$$\begin{aligned}\eta_2 = {} & 48(1 - c_3)\gamma^3 + 8(3c_3^2 + 3c_3 - 5)\gamma^2 \\ & + 6(-4c_3^2 + 2c_3 + 1)\gamma + 4c_3^2 - 3c_3\end{aligned} \tag{33b}$$

Table 6 Order condition expressed using $\boldsymbol{q}^{(k)}$ for an ESDIRK

$\sigma_j^{(i)}$	Expression	$\sigma_j^{(i)}$	Expression
$\sigma_1^{(1)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{e}-1$	$\sigma_1^{(2)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{e}-\frac{1}{2}$
$\sigma_1^{(3)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^2\boldsymbol{e}-\frac{1}{6}$	$\sigma_2^{(3)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{q}^{(2)}+\sigma_1^{(3)}$
$\sigma_1^{(4)}$	$\frac{1}{6}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^3\boldsymbol{e}-\frac{1}{24}$	$\sigma_2^{(4)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{q}^{(2)}+3\sigma_1^{(4)}$
$\sigma_3^{(4)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{q}^{(3)}+\sigma_1^{(4)}$	$\sigma_4^{(4)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}^{(2)}+\sigma_3^{(4)}$
$\sigma_1^{(5)}$	$\frac{1}{24}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^4\boldsymbol{e}-\frac{1}{120}$	$\sigma_2^{(5)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^2\boldsymbol{q}^{(2)}+6\sigma_1^{(5)}$
$\sigma_3^{(5)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\left(\boldsymbol{Q}^{(2)}+\boldsymbol{C}^2\right)\boldsymbol{q}^{(2)}+3\sigma_1^{(5)}$	$\sigma_4^{(5)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{q}^{(3)}+4\sigma_1^{(5)}$
$\sigma_5^{(5)}$	$\frac{1}{6}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{q}^{(4)}+\sigma_1^{(5)}$	$\sigma_6^{(5)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{A}\boldsymbol{q}^{(2)}+\sigma_4^{(5)}$
$\sigma_7^{(5)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{C}\boldsymbol{q}^{(2)}+3\sigma_5^{(5)}$	$\sigma_8^{(5)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}^{(3)}+\sigma_5^{(5)}$
$\sigma_9^{(5)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}^2\boldsymbol{q}^{(2)}+\sigma_8^{(5)}$	$\sigma_1^{(6)}$	$\frac{1}{120}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^5\boldsymbol{e}-\frac{1}{720}$
$\sigma_2^{(6)}$	$\frac{1}{6}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^3\boldsymbol{q}^{(2)}+10\sigma_1^{(6)}$	$\sigma_3^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\left(\boldsymbol{Q}^{(2)}+\boldsymbol{C}^2\right)\boldsymbol{q}^{(2)}+15\sigma_1^{(6)}$
$\sigma_4^{(6)}$	$\frac{1}{4}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^2\boldsymbol{q}^{(3)}+10\sigma_1^{(6)}$	$\sigma_5^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\left(\boldsymbol{Q}^{(3)}+\frac{1}{3}\boldsymbol{C}^3\right)\boldsymbol{q}^{(2)}+\sigma_4^{(6)}$
$\sigma_6^{(6)}$	$\frac{1}{6}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{q}^{(4)}+5\sigma_1^{(6)}$	$\sigma_7^{(6)}$	$\frac{1}{24}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{q}^{(5)}+\sigma_1^{(6)}$
$\sigma_8^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^2\boldsymbol{A}\boldsymbol{q}^{(2)}+\sigma_4^{(6)}$	$\sigma_9^{(6)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{Q}^{(2)}\boldsymbol{A}\boldsymbol{q}^{(2)}+\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{Q}^{(2)}\boldsymbol{q}^{(3)}+\frac{1}{6}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}^3\boldsymbol{q}^{(2)}+\sigma_8^{(6)}$
$\sigma_{10}^{(6)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{A}\boldsymbol{C}\boldsymbol{q}^{(2)}+3\sigma_6^{(6)}$	$\sigma_{11}^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{C}^2\boldsymbol{q}^{(2)}+6\sigma_7^{(6)}$
$\sigma_{12}^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\left(\boldsymbol{Q}^{(2)}+\boldsymbol{C}^2\right)\boldsymbol{q}^{(2)}+3\sigma_7^{(6)}$	$\sigma_{13}^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{A}\boldsymbol{q}^{(3)}+\sigma_6^{(6)}$
$\sigma_{14}^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{C}\boldsymbol{q}^{(3)}+4\sigma_7^{(6)}$	$\sigma_{15}^{(6)}$	$\frac{1}{6}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}^{(4)}+\sigma_7^{(6)}$
$\sigma_{16}^{(6)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{A}^2\boldsymbol{q}^{(2)}+\sigma_{13}^{(6)}$	$\sigma_{17}^{(6)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{C}\boldsymbol{A}\boldsymbol{q}^{(2)}+\sigma_{14}^{(6)}$
$\sigma_{18}^{(6)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}^2\boldsymbol{C}\boldsymbol{q}^{(2)}+3\sigma_{15}^{(6)}$	$\sigma_{19}^{(6)}$	$\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}^2\boldsymbol{q}^{(3)}+\sigma_{15}^{(6)}$
$\sigma_{20}^{(6)}$	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}^3\boldsymbol{q}^{(2)}+\sigma_{19}^{(6)}$		

Then γ has been given to offer the tunable algorithmic dissipation, and c_3, c_4 are obtained by minimizing $||\boldsymbol{\sigma}^{(5)}||_2$ for several fixed ρ_∞, as listed in Table 7.

For MSSTH(5), the order conditions impose

$$c_2 = 2\gamma \tag{34a}$$

$$a_{21} = \gamma \tag{34b}$$

$$a_{32} = \frac{c_3(c_3-2\gamma)}{4\gamma} \tag{34c}$$

$$a_{31} = c_3 - \gamma - a_{32} \tag{34d}$$

$$a_{42} = -\frac{c_4(c_4 - 2\gamma)\eta_1}{4\gamma(c_3 - 2\gamma)\eta_2} \tag{34e}$$

$$a_{43} = \frac{c_4^2 - 4a_{42}\gamma - 2c_4\gamma}{2c_3} \tag{34f}$$

$$a_{41} = c_4 - \gamma - a_{42} - a_{43} \tag{34g}$$

$$c_5 = \frac{\eta_3}{\eta_4} \tag{34h}$$

$$a_{53} = \frac{c_5(c_5 - c_3)(c_5 - 2\gamma)\eta_5}{c_3(c_4 - c_3)(c_3 - 2\gamma)\eta_6} \tag{34i}$$

$$a_{54} = \frac{c_5(c_5 - c_3)(c_5 - c_4)(c_5 - 2\gamma)\eta_7}{c_4(c_4 - c_3)(c_4 - 2\gamma)\eta_6} \tag{34j}$$

$$a_{52} = \frac{c_5^2 - 2a_{53}c_3 - 2a_{54}c_4 - 2c_5\gamma}{4\gamma} \tag{34k}$$

$$a_{51} = c_5 - \gamma - a_{52} - a_{53} - a_{54} \tag{34l}$$

$$b_3 = -\frac{\eta_8}{60c_3(c_4 - c_3)(c_5 - c_3)(c_3 - 2\gamma)} \tag{34m}$$

$$b_4 = \frac{\eta_9}{60c_4(c_4 - c_3)(c_5 - c_4)(c_4 - 2\gamma)} \tag{34n}$$

$$b_5 = -\frac{\eta_{10}}{60c_5(c_5 - c_3)(c_5 - c_4)(c_5 - 2\gamma)} \tag{34o}$$

$$b_2 = \frac{1 - 2\gamma - 2b_3c_3 - 2b_4c_4 - 2b_5c_5}{4\gamma} \tag{34p}$$

$$b_1 = 1 - \gamma - b_2 - b_3 - b_4 - b_5 \tag{34q}$$

with

$$\begin{aligned}\eta_1 = {} & 240(1 - c_4)\gamma^4 + 40(3c_3^2 - 6c_3 + 12c_4 - 7)\gamma^3 \\ & + 20(-9c_3^2 + 13c_3 - 12c_4 + 4)\gamma^2 \\ & + (60c_3^2 - 70c_3 + 40c_4 - 6)\gamma - 5c_3^2 + 5c_3 - 2c_4\end{aligned} \tag{35a}$$

$$\begin{aligned}\eta_2 = {} & 120(1 - c_3)\gamma^3 + 20(9c_3 - 7)\gamma^2 \\ & + 20(2 - 3c_3)\gamma + 5c_3 - 3\end{aligned} \tag{35b}$$

$$\begin{aligned}\eta_3 = {} & -720\gamma^5 + (360c_3 + 848)\gamma^4 \\ & + (-336c_3 - 368)\gamma^3 + (120c_3 + 64)\gamma^2 \\ & + (-18c_3 - 4)\gamma + c_3\end{aligned} \tag{35c}$$

$$\begin{aligned}\eta_4 = {} & -480\gamma^5 + (240c_3 + 600)\gamma^4 \\ & + (-240c_3 - 288)\gamma^3 + (96c_3 + 56)\gamma^2\end{aligned} \tag{35d}$$

Table 7 Parameters used in the modified MSSTH(n) for several fixed ρ_∞

ρ_∞	MSSTH(3): γ	MSSTH(4): γ	MSSTH(4): c_3
0.0	0.18042530642939832996596 29	0.57281606248213501331179 03	0.55909857542294173051525 42
0.1	0.17861945820465807699406 47	0.54836664497582987554125 11	0.60029388886983241219751 47
0.2	0.17694580661822370548641 46	0.52638645684238627442397 27	0.63852281448916059533286 10
0.3	0.17538551584284575723948 76	0.50633011897078195051591 36	0.67524540713318077522649 00
0.4	0.17392360787719712833521 16	0.48779747481234808637040 60	0.71166263135355824400370 08
0.5	0.17254796142208930764816 44	0.47048057762167683204523 88	0.74893739013168579457010 66
0.6	0.17124861859069104297326 19	0.45413078503652870576701 16	0.78843701152100361551195 26
0.7	0.17001729177247645874437 86	0.43853618990219250806106 29	0.83214959599683093609817 58
0.8	0.16884700467916768928944 29	0.42350376606717887728592 59	0.88365850394190859053364 49
0.9	0.16773182575688677653502 62	0.40884186612069933763891 07	0.95088331353432272533372 52
1.0	0.16666666666666666666666 67	0.39433756729740654378701 95	1.05348034117028673017024 00

ρ_∞	MSSTH(4): c_4	MSSTH(5): γ	MSSTH(5): c_4
0.0	0.74140116648336540361441 39	0.27805384113644993071545 74	0.96732586055716962558648 22
0.1	0.75841298757803721208858 86	0.27414130603186848134100 73	0.90855001848651739670970 07
0.2	0.77314366596046124602281 68	0.27045988677458177029677 70	0.85109126740887963702419 94
0.3	0.78603120641227375298143 44	0.26697804392565055442432 25	0.79517804197092967211091 26
0.4	0.79725148192031436433779 85	0.26367023171160552941216 79	0.74096898647213860211735 44
0.5	0.80671407474270417914397 06	0.26051541660705490599525 55	0.68856097870408505823291 99
0.6	0.81396625294191349286876 40	0.25749602985667474630560 04	0.63799727905607228617412 83
0.7	0.81793010320067149887535 15	0.25459720817013348215240 85	0.58927567838046857051637 06
0.8	0.81624789465002423050066 23	0.25180623118384964920224 43	0.54235622095961477651115 96
0.9	0.80362953525688307632179 89	0.24911209652962970628742 31	0.49716834141991045533715 01
1.0	0.76893053626170526637650 94	0.24650519314282015592709 74	0.45361725766875554688439 82

$$
\begin{aligned}
&+ (-16c_3 - 4)\gamma + c_3 \\
\eta_5 &= 120(-c_4^2 + 2c_4 - c_3 - c_5 + c_3c_5)\gamma^3 \qquad (35e)\\
&+ 20(9c_4^2 - 15c_4 + 8c_3 + 7c_5 - 9c_3c_5)\gamma^2 \\
&+ 10(-6c_4^2 + 9c_4 - 5c_3 - 4c_5 + 6c_3c_5)\gamma \\
&+ 5c_4^2 - 7c_4 + 4c_3 + 3c_5 - 5c_3c_5 \\
\eta_6 &= 120(c_3 + c_4 - c_3c_4 - 1)\gamma^2 \qquad (35f)\\
&+ 10(12c_3c_4 - 10c_4 - 10c_3 + 9)\gamma \\
&+ 15c_3 + 15c_4 - 20c_3c_4 - 12 \\
\eta_7 &= 120(1 - c_3)\gamma^3 + 20(9c_3 - 7)\gamma^2 \qquad (35g)\\
&+ 20(2 - 3c_3)\gamma + 5c_3 - 3 \\
\eta_8 &= 120(c_4 + c_5 - c_4c_5 - 1)\gamma^2 \qquad (35h)\\
&+ 10(12c_4c_5 - 10c_5 - 10c_4 + 9)\gamma
\end{aligned}
$$

$$
\begin{aligned}
&+ 15c_4 + 15c_5 - 20c_4c_5 - 12 \\
\eta_9 = {} & 120(c_3 + c_5 - c_3c_5 - 1)\gamma^2 \qquad (35i) \\
&+ 10(12c_3c_5 - 10c_5 - 10c_3 + 9)\gamma \\
&+ 15c_3 + 15c_5 - 20c_3c_5 - 12 \\
\eta_{10} = {} & 120(c_3 + c_4 - c_3c_4 - 1)\gamma^2 \qquad (35j) \\
&+ 10(12c_3c_4 - 10c_4 - 10c_3 + 9)\gamma \\
&+ 15c_3 + 15c_4 - 20c_3c_4 - 12
\end{aligned}
$$

leaving γ, c_3 and c_4 free. Here we assume $c_3 = 0.1$, and then c_4 is obtained by minimizing $||\boldsymbol{\sigma}^{(6)}||_2$. The values used for γ and c_4 are listed in Table 7.

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

Declarations

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

References

1. Alexander, R.: Diagonally implicit Runge–Kutta methods for stiff ODEs. SIAM J. Numer. Anal. **14**(6), 1006–1021 (1977)
2. Arnold, M., Brüls, O.: Convergence of the generalized-α scheme for constrained mechanical systems. Multibody Syst. Dyn. **18**(2), 185–202 (2007)
3. Ascher, U.M., Petzold, L.R.: Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, vol. 61. SIAM, Philadelphia (1998)
4. Bauchau, O.A., Betsch, P., Cardona, A., Gerstmayr, J., Jonker, B., Masarati, P., Sonneville, V.: Validation of flexible multibody dynamics beam formulations using benchmark problems. Multibody Syst. Dyn. **37**(1), 29–48 (2016)
5. 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
6. Boom, P.D., Zingg, D.W.: Optimization of high-order diagonally-implicit Runge–Kutta methods. J. Comput. Phys. **371**, 168–191 (2018)
7. Brenan, K.E., Campbell, S.L.V., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. North-Holland, New York (1989)
8. Brüls, O., Arnold, M.: The generalized-α scheme as a linear multistep integrator: toward a general mechatronic simulator. J. Comput. Nonlinear Dyn. **3**(4), 041007 (2008). https://doi.org/10.1115/1.2960475
9. Butcher, J.C.: Implicit Runge–Kutta processes. Math. Comput. **18**(85), 50–64 (1964)
10. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. Wiley, New York (2016)
11. Chung, J., Hulbert, G.: A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. J. Appl. Mech. **60**, 371–375 (1993)

12. Dahlquist, G.G.: A special stability problem for linear multistep methods. BIT Numer. Math. **3**(1), 27–43 (1963)
13. de Alwis, T.: Central difference formula in numerical analysis. PRIMUS, Probl. Resour. Issues Math. Undergrad. Stud. **2**(2), 165–172 (1992)
14. Faltinsen, S., Marthinsen, A., Munthe-Kaas, H.Z.: Multistep methods integrating ordinary differential equations on manifolds. Appl. Numer. Math. **39**(3), 349–365 (2001). https://doi.org/10.1016/S0168-9274(01)00103-9
15. Ghiringhelli, G.L., Masarati, P., Mantegazza, P.: Multibody implementation of finite volume C^0 beams. AIAA J. **38**(1), 131–138 (2000)
16. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations, II, Stiff and Differential-Algebraic Problems. Springer, Berlin (1991)
17. 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)
18. Hoff, C., Pahl, P.: Development of an implicit method with numerical dissipation from a generalized single-step algorithm for structural dynamics. Comput. Methods Appl. Mech. Eng. **67**(3), 367–385 (1988)
19. Houbolt, J.C.: A recurrence matrix solution for the dynamic response of elastic aircraft. J. Aeronaut. Sci. **17**(9), 540–550 (1950)
20. Jansen, K.E., Whiting, C.H., Hulbert, G.M.: A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. Comput. Methods Appl. Mech. Eng. **190**(3–4), 305–319 (2000). https://doi.org/10.1016/S0045-7825(00)00203-6
21. Kennedy, C.A., Carpenter, M.H.: Diagonally implicit Runge–Kutta methods for stiff ODEs. Appl. Numer. Math. **146**, 221–244 (2019)
22. Köbis, M., Arnold, M.: Convergence of generalized-α time integration for nonlinear systems with stiff potential forces. Multibody Syst. Dyn. **37**(1), 107–125 (2016)
23. Masarati, P., Lanz, M., Mantegazza, P.: Multistep integration of ordinary, stiff and differential-algebraic problems for multibody dinamics applications. In: XVI Congresso Nazionale AIDAA, pp. 1–10 (2001)
24. Masarati, P., Morandini, M., Mantegazza, P.: An efficient formulation for general-purpose multibody/multiphysics analysis. J. Comput. Nonlinear Dyn. **9**(4), 041001 (2014)
25. Newmark, N.M.: A method of computation for structural dynamics. J. Eng. Mech. Div. **85**(3), 67–94 (1959)
26. Noh, G., Bathe, K.J.: The Bathe time integration method with controllable spectral radius: the ρ_∞-Bathe method. Comput. Struct. **212**, 299–310 (2019)
27. Owren, B., Simonsen, H.H.: Alternative integration methods for problems in structural dynamics. Comput. Methods Appl. Mech. Eng. **122**(1–2), 1–10 (1995). https://doi.org/10.1016/0045-7825(94)00717-2
28. Park, K.: An improved stiffly stable method for direct integration of nonlinear structural dynamic equations. J. Appl. Mech. **42**(3), 464–470 (1975)
29. Schiehlen, W., et al.: Multibody Systems Handbook, vol. 6. Springer, Berlin (1990)
30. Tamma, K.K., Har, J., Zhou, X., Shimada, M., Hoitink, 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)
31. Wilson, E.L.: A computer program for the dynamic stress analysis of underground structures. Tech. rep., California Univ Berkeley Structural Engineering Lab (1968)
32. Wood, W., Bossak, M., Zienkiewicz, O.: An alpha modification of Newmark's method. Int. J. Numer. Methods Eng. **15**(10), 1562–1566 (1980)
33. 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)
34. 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)
35. Zhang, H., Zhang, R., Xing, Y., Masarati, P.: On the optimization of n-sub-step composite time integration methods. Nonlinear Dyn. **102**(3), 1939–1962 (2020)
36. Zhang, H., Zhang, R., Masarati, P.: Improved second-order unconditionally stable schemes of linear multi-step and equivalent single-step integration methods. Comput. Mech. **67**(1), 289–313 (2021). https://doi.org/10.1007/s00466-020-01933-y
37. 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)