Appl. Math. Mech. -Engl. Ed., **38**(6), 889–908 (2017)
DOI 10.1007/s10483-017-2207-8

Applied Mathematics and Mechanics (English Edition)

An improved time integration scheme based on uniform cubic B-splines and its application in structural dynamics*

Weibin WEN[1], Hongshuai LEI[2], Kai WEI[3,†], Baosheng XU[4], Shengyu DUAN[2], Daining FANG[1,2]

1. College of Engineering, Peking University, Beijing 100871, China;
2. Beijing Institute of Technology, Beijing 100081, China;
3. State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, School of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China;
4. Key Laboratory of Applied Mechanics, Department of Engineering Mechanics, School of Aerospace Engineering, Tsinghua University, Beijing 100084, China

Abstract A time integration algorithm for structural dynamic analysis is proposed by uniform cubic B-spline functions. The proposed algorithm is successfully used to solve the dynamic response of a single degree of freedom (SDOF) system, and then is generalized for a multiple-degree of freedom (MDOF) system. Stability analysis shows that, with an adjustable algorithmic parameter, the proposed method can achieve both conditional and unconditional stabilities. Validity of the method is shown with four numerical simulations. Comparison between the proposed method and other methods shows that the proposed method possesses high computation accuracy and desirable computation efficiency.

Key words dynamical system, time integration, stability, dynamic response, B-spline

Chinese Library Classification O347
2010 Mathematics Subject Classification 74Kxx, 34N05

1 Introduction

During the last several decades, a large number of researches have been conducted on the exploration of efficient time integration schemes for the linear and nonlinear analysis of structures[1–4]. Generally, there are two major categories of time integration methods, namely, the explicit method[5–7] and the implicit method[8–10].

The implicit method is popular for a practical use because it can give a stable solution in the case of linear system, and the time increment can be easily selected. In the case of nonlinear analysis, it may be necessary for implicit schemes to update the stiffness matrix and conduct the iteration calculation within each time increment. Consequently, a lot of time is occupied for matrix inversion and iteration calculation. Meanwhile, a numerical divergence may be easily induced when a strong nonlinearity exists in a system[11]. Therefore, various explicit schemes have been designed to overcome the above disadvantages of the implicit method. Compared

* Received May 3, 2016 / Revised Sept. 27, 2016
Project supported by the National Natural Science Foundation of China (Nos. 11602004 and 11602081) and the Fundamental Research Funds for the Central Universities (No. 531107040934)
† Corresponding author, E-mail: weikai@hnu.edu.cn

with implicit methods, explicit methods need no effort for iteration calculation in each time step. Thus, fewer storage and computational efforts are required for explicit methods[5]. The shortcoming of the explicit method is that almost all explicit methods are conditionally stable. Consequently, a small time increment and relatively large time steps may be required in a time history analysis, which, to a great extent, lowers the computation efficiency. However, a very small time increment can decrease the linearization errors for nonlinear systems. Actually, a small time increment can effectively capture intermediate and high frequency modes with desirable accuracy. Therefore, the explicit method is very suitable for the cases such as wave propagation and shock response problems where intermediate and high frequency modes are very significant for dynamic responses[12–14].

Recently, Bathe and Noh[9] developed a simple implicit time integration method. This method can remain unconditionally stable, without the use of adjustable parameters. Then, Noh and Bathe[15] presented an explicit time integration scheme for the solution of wave propagation problems. The presented method possesses the second-order accuracy for different dynamic systems. A clear advantage of this method is that it is very effective in suppressing or filtering out spurious high-frequency modes. However, for the above two methods, each time step consists of two sub-steps, and thus requires much larger computation efforts than other conventional methods.

Recently, the B-spline method, as a powerful numerical tool, was used to develop a novel explicit time integration method by Rostami et al.[16]. Compared with other conventional schemes, the presented B-spline method possesses higher computation efficiency. However, it is conditionally stable.

Wen et al.[17–18] developed an explicit time integration method using septuple and quintic B-splines. With three adjustable algorithmic parameters, the presented methods can achieve both conditional and unconditional stability. Although the proposed method possesses higher computation accuracy, much more computation time is consumed, especially for the dynamic system with high degrees of freedom.

In this study, a new method is thus proposed by use of a family of uniform cubic B-spline functions. The paper is organized as follows. In Section 2, we introduce the explicit definition of B-spline and the expression of cubic B-spline interpolation. Then, the cubic B-spline interpolation is used to solve the equilibrium equation of linear single degree of freedom (SDOF) system in Section 3. Subsequently, the implementation of cubic B-spline interpolation on the multiple-degree of freedom (MDOF) system coupled with its calculation procedure is given in Section 4. The stability and accuracy analyses are conducted in Section 5. The validity of the proposed method is verified with four numerical examples in Section 6. Finally, in Section 7, we draw the conclusions of this study.

2 Overview of cubic B-splines

2.1 Explicit definition of B-spline function

There are many ways to define B-spline functions[17,19]. To simplify computer implementation, here we adopt an explicit definition to calculate B-spline functions[19]. Thus, the ith B-spline basis function of the dth degree $B_{i,d}(t)$, briefly denoted by $B_{i,d}$, can be defined by

$$B_{i,0}(t) = \begin{cases} 1, & t_i \leqslant t \leqslant t_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

$$B_{i,d}(t) = \Big(\frac{t - t_i}{t_{d+i} - t_i}\Big) B_{i,d-1}(t) + \Big(\frac{t_{d+i+1} - t}{t_{d+i+1} - t_{i+1}}\Big) B_{i+1,d-1}(t), \tag{2}$$

where the knot span $\{t_0, t_1, \cdots, t_i, t_{i+1}, \cdots, t_{d+i}, t_{d+i+1}, \cdots\}$ must be a non-decreasing sequence of real numbers, that is, $t_i \leqslant t_{i+1}$. Because Eq. (2) can yield the quotient $0/0$, we define this

quotient to be zero. Without going into details, $B_{i,d}(t)$ are linear combinations of two $(d-1)$-degree B-spline functions.

$B_{i,d}(t)$ are non-negative and local supported over the interval $[t_i, t_{i+d+1}]$. As shown in Fig. 1, for cubic B-spline basis functions used in this study, the usable parameter range of $B_{i,3}(t)$ is $[t_i, t_{i+4}]$. Besides, we have $\sum_{i=-\infty}^{\infty} B_{i,d}(t) = 1$ within the usable interpolation domain $[t_i, t_{i+1}]$. Clearly, when $t_i < t_{i+1}$, $B_{i,d}(t)$ are $(d-1)$ times continuously differentiable.

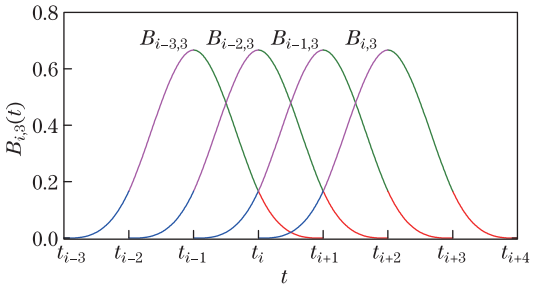


Fig. 1 Cubic B-spline basis functions, where usable interpolation range is from t_i to t_{i+1}

2.2 Preliminary results of cubic B-spline interpolation

In this study, we use piecewise polynomials for the subsequent analysis. Let the time domain $[a, b]$ be divided into n subintervals $I_i \equiv [t_i, t_{i+1}]\,(i = 0, 1, \cdots, n-1)$ by a set of equidistant knots $t_i = a + i\Delta t$, where $\Delta t = h = \frac{b-a}{n}$. As shown in Fig. 1, to construct the cubic B-spline interpolation functions within the whole time domain $[a, b]$, we should extend the set of knots as $t_{-3} < \cdots < t_{-1} < a = t_0 < t_1 < \cdots < t_n = b < \cdots < t_{n+3}$.

For any knot $t_j\,(j = -3, -2, \cdots, t_{n+3})$, we have $t_j = t_i + (j-i)h$. Thus, $B_{i,3}(t)\,(i = 0, 1, \cdots, n-1)$ can be simplified as

$$B_{i,3}(t) = \frac{1}{3!h^3}\begin{cases} (t-t_i)^3, & t \in [t_i, t_{i+1}], \\ (t-t_i)^3 - 4(t-t_{i+1})^3, & t \in [t_{i+1}, t_{i+2}], \\ (t_{i+4}-t)^3 - 4(t_{i+3}-t)^3, & t \in [t_{i+2}, t_{i+3}], \\ (t_{i+4}-t)^3, & t \in [t_{i+3}, t_{i+4}]. \end{cases} \tag{3}$$

Equation (3) shows that $B_{j,3}(t)$ are the shifted instances of $B_{i,3}(t)$, that is, $B_{j,3}(t) = B_{i,3}(t - (j-i)h)$. For any usable interpolation subinterval $I_i \equiv [t_i, t_{i+1}]$ in Fig. 1, only four piecewise cubic B-spline functions are contributory to the B-spline interpolation.

In this study, the extended subintervals $I_i' \equiv [t_i, t_{i+\varphi}]\,(1 \leqslant \varphi < 2, i = 0, 1, \cdots, n-1)$ are used for the analysis. More specifically, the cubic B-spline interpolation function over any subinterval I_i' is used to approximate the exact solution within the subintervals I_i.

Further, let $\tau_i = (t - t_i)/(\varphi\Delta t)$ (i.e., $t = t_i + \tau_i\varphi\Delta t, 1 < \varphi < 2$), the cubic B-spline functions in Fig. 2 and their lth derivatives with respect to the variable t can be expressed as

$$B_{-3,3}^{(l)}(\tau_i) = \frac{(-\varphi\Delta t)^{-l}}{(3-l)!}(1-\tau_i)^{3-l}, \tag{4}$$

$$B_{-2,3}^{(l)}(\tau_i) = \frac{(-\varphi\Delta t)^{-l}}{(3-l)!}((2-\tau_i)^{3-l} - 4(1-\tau_i)^{3-l}), \tag{5}$$

$$B_{-1,3}^{(l)}(\tau_i) = \frac{(\varphi\Delta t)^{-l}}{(3-l)!}((\tau_i+1)^{3-l} - 4(\tau_i)^{3-l}), \tag{6}$$

$$B_{0,3}^{(l)}(\tau_i) = \frac{(\varphi\Delta t)^{-l}}{(3-l)!}(\tau_i)^{3-l}, \tag{7}$$

where $l = 0, 1, 2$.

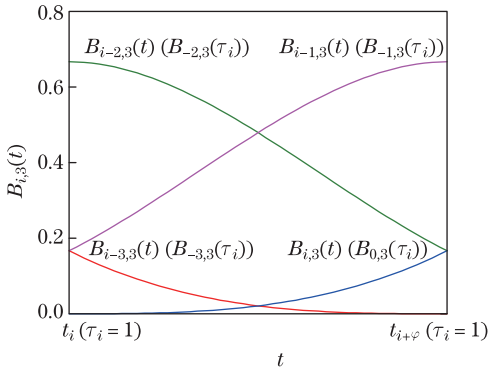


Fig. 2 Usable piecewise cubic B-splines within time subintervals $I_i' \equiv [t_i, t_{i+\varphi}]$

Thus, the cubic B-spline interpolation function within any subinterval I_i' can be defined by

$$S_i(t) = \sum_{k=-3}^{0} C_k^i B_{k,3}(\tau_i), \quad i = 0, 1, \cdots, n-1, \tag{8}$$

where C_k^i are unknown real coefficients.

With Eqs. (4)–(7), Eq. (8) can be extended as

$$S_i^{(l)}(t) = \boldsymbol{B}^{(l)}(\tau_i)\boldsymbol{C}_i, \quad \forall t \in I_i'(t), \quad \forall \tau_i \in [0, 1], \quad \varphi \in [1, 2), \quad l = 0, 1, 2, \tag{9}$$

where

$$\boldsymbol{B}^{(l)}(\tau_i) = [B_{-3,3}^{(l)}(\tau_i) \quad B_{-2,3}^{(l)}(\tau_i) \quad B_{-1,3}^{(l)}(\tau_i) \quad B_{0,3}^{(l)}(\tau_i)], \tag{10}$$

$$\boldsymbol{C}_i = [C_{-3}^i \quad C_{-2}^i \quad C_{-1}^i \quad C_0^i]^{\mathrm{T}}. \tag{11}$$

Clearly, for $t = t_i$, $t = t_{i+\varphi}$, and $t = t_{i+1}$, we have, respectively,

$$S_i^{(l)}(t_i) = \boldsymbol{B}^{(l)}(0)\boldsymbol{C}_i, \tag{12}$$

$$S_i^{(l)}(t_{i+\varphi}) = \boldsymbol{B}^{(l)}(1)\boldsymbol{C}_i, \tag{13}$$

and

$$S_i^{(l)}(t_{i+1}) = \boldsymbol{B}^{(l)}\Big(\frac{1}{\varphi}\Big)\boldsymbol{C}_i. \tag{14}$$

To solve $\boldsymbol{B}^{(l)}(0)$ and $\boldsymbol{B}^{(l)}(1)$, we tabulate $B_{i,3}^{(l)}(\tau_i)$ $(i = -3, -2, -1, 0, l = 0, 1, 2)$ that evaluate at $\tau_i = 0$ (i.e., $t = t_i$) and $\tau_i = 1$ (i.e., $t = t_{i+\varphi}$) in Table 1.

Table 1 Values of $B_{i,3}^{(l)}(0)$ and $B_{i,3}^{(l)}(1)$

l	$B_{-3,3}^{(l)}(0)$	$B_{-2,3}^{(l)}(0)$	$B_{-1,3}^{(l)}(0)$	$B_{0,3}^{(l)}(0)$	$B_{-3,3}^{(l)}(1)$	$B_{-2,3}^{(l)}(1)$	$B_{-1,3}^{(l)}(1)$	$B_{0,3}^{(l)}(1)$
0	$1/6$	$2/3$	$1/6$	0	0	$1/6$	$2/3$	$1/6$
1	$-1/(2\varphi\Delta t)$	0	$1/(2\varphi\Delta t)$	0	0	$-1/(2\varphi\Delta t)$	0	$1/(2\varphi\Delta t)$
2	$1/(\varphi^2(\Delta t)^2)$	$-2/(\varphi^2(\Delta t)^2)$	$1/(\varphi^2(\Delta t)^2)$	0	0	$1/(\varphi^2(\Delta t)^2)$	$-2/(\varphi^2(\Delta t)^2)$	$1/(\varphi^2(\Delta t)^2)$

3 Implementation of cubic B-spline on SDOF systems

For an SDOF system, a linear differential equation of motion can be written as

$$u^{(2)}(t) + 2\xi\omega u^{(1)}(t) + \omega^2 u(t) = f(t), \tag{15}$$

where ξ, ω, and $f(t)$ are the damping ratio, the undamped circular natural frequency, and the modal forcing excitation of the system, respectively. $u(t)$, $u^{(1)}(t)$, and $u^{(2)}(t)$ are the displacement, velocity, and acceleration functions, respectively.

The initial value problem is to solve Eq. (15) to meet the given initial conditions of $u(t_0) = a_0$, $u^{(1)}(t_0) = a_1$, and $u^{(2)}(t_0) = a_2$.

Let t_0, t_1, t_2, $\cdots$, t_{n-1} be n time knots over the whole time interval $[0, t_f]$ so that $t_i = i\Delta t, i = 0, 1, 2, \cdots, n-1$, $t_0 = 0$, and $t_{n-1} = t_f$. Using Eqs. (8) and (9), the approximate solution to Eq. (15) and its lth derivatives, within any subinterval I_i' $(i = 0, 1, 2, \cdots, n-1)$, can be expressed as

$$u_i^{(l)}(t) = \sum_{k=-3}^{0} C_k^i B_{k,3}^{(l)}(\tau_i), \quad l = 0, 1, 2, \tag{16}$$

where the subscript i of $u_i(t)$ denotes the number of time subinterval.

To simplify the subsequent deduction, we let

$$\widetilde{u}_i^{(l)}(t) = (\varphi\Delta t)^{(l)} \cdot u_i^{(l)}(t), \quad l = 0, 1, 2, \tag{17}$$

$$\widetilde{B}_{k,5}^{(l)}(\tau_i) = (\varphi\Delta t)^{(l)} \cdot B_{k,3}^{(l)}(\tau_i), \quad l = 0, 1, 2. \tag{18}$$

Substituting $t = t_i$ (i.e., $\tau_i = 0$) and $t = t_{i+\varphi}$ (i.e., $\tau_i = 1$) into Eq. (16) begets six equations. With Eqs. (17) and (18) and Table 1, these six equations can be expressed in the explicit forms of

$$\widetilde{u}_i(t_i) = \frac{1}{6}C_{-3}^i + \frac{2}{3}C_{-2}^i + \frac{1}{6}C_{-1}^i, \tag{19a}$$

$$\widetilde{u}_i^{(1)}(t_i) = -\frac{1}{2}C_{-3}^i + 0C_{-2}^i + \frac{1}{2}C_{-1}^i, \tag{19b}$$

$$\widetilde{u}_i^{(2)}(t_i) = C_{-3}^i - 2C_{-2}^i + C_{-1}^i, \tag{19c}$$

and

$$\widetilde{u}_i(t_{i+\varphi}) = \frac{1}{6}C_{-2}^i + \frac{2}{3}C_{-1}^i + \frac{1}{6}C_0^i, \tag{20a}$$

$$\widetilde{u}_i^{(1)}(t_{i+\varphi}) = -\frac{1}{2}C_{-2}^i + 0C_{-1}^i + \frac{1}{2}C_0^i, \tag{20b}$$

$$\widetilde{u}_i^{(2)}(t_{i+\varphi}) = C_{-2}^i - 2C_{-1}^i + C_0^i. \tag{20c}$$

Using Eqs. (19a)–(19c), C_k^i $(k = -3, -2, -1, i = 0, 1, 2, \cdots, n-1)$ can be calculated by

$$C_{-3}^i = \widetilde{u}_i(t_i) - \widetilde{u}_i^{(1)}(t_i) + \frac{1}{3}\widetilde{u}_i^{(2)}(t_i), \tag{21a}$$

$$C_{-2}^i = \widetilde{u}_i(t_i) + 0\widetilde{u}_i^{(1)}(t_i) - \frac{1}{6}\widetilde{u}_i^{(2)}(t_i), \tag{21b}$$

$$C_{-1}^i = \widetilde{u}_i(t_i) + \widetilde{u}_i^{(1)}(t_i) + \frac{1}{3}\widetilde{u}_i^{(2)}(t_i). \tag{21c}$$

Prescribing that Eq. (15) is satisfied at $t = t_{i+\varphi}$ (i.e., $\tau_i = 1$) renders

$$u_i^{(2)}(t_{i+\varphi}) + 2\xi\omega u_i^{(1)}(t_{i+\varphi}) + \omega^2 u_i(t_{i+\varphi}) = f(t_{i+\varphi}), \quad i = 0, 1, 2, \cdots, n-1. \tag{22}$$

With Eq. (17), Eq. (22) can be written as

$$\widetilde{u}_i^{(2)}(t_{i+\varphi}) + 2\xi\varphi \cdot \omega\Delta t \cdot \widetilde{u}_i^{(1)}(t_{i+\varphi}) + \varphi^2 \cdot (\omega\Delta t)^2 \cdot \widetilde{u}_i(t_{i+\varphi}) = (\varphi\Delta t)^2 \cdot f(t_{i+\varphi}). \tag{23}$$

Substituting Eqs. (20a)–(20c) into Eq. (23) yields

$$\alpha C_{-2}^i + \beta C_{-1}^i + \gamma C_0^i = (\varphi\Delta t)^2 f(t_{i+\varphi}), \quad i = 0, 1, 2, \cdots, n-1, \tag{24}$$

where α, β, and γ are given by

$$\alpha = \frac{1}{6}\varphi^2 \cdot (\omega\Delta t)^2 - \xi\varphi\omega\Delta t + 1, \tag{25a}$$

$$\beta = \frac{2}{3}\varphi^2 \cdot (\omega\Delta t)^2 - 2, \tag{25b}$$

$$\gamma = \frac{1}{6}\varphi^2 \cdot (\omega\Delta t)^2 + \xi\varphi\omega\Delta t + 1. \tag{25c}$$

C_{-3}^i, C_{-2}^i, and C_{-1}^i can be previously solved by Eqs. (21a)–(21c). Therefore, with Eq. (24), we can solve C_0^i as

$$C_0^i = \frac{1}{\gamma}((\varphi\Delta t)^2 f(t_{i+\varphi}) - \alpha C_{-2}^i - \beta C_{-1}^i). \tag{26}$$

After solving C_{-3}^i, C_{-2}^i, C_{-1}^i, and C_0^i, we can use Eqs. (16)–(18) to determine $\widetilde{u}_i^{(l)}(t_{i+1})$ as

$$\widetilde{u}_i^{(l)}(t_{i+1}) = \sum_{k=-3}^{0} C_k^i \widetilde{B}_{k,3}^{(l)}\Big(\frac{1}{\varphi}\Big), \quad l = 0, 1, 2, \quad i = 0, 1, 2, \cdots, n-1. \tag{27}$$

After solving $\widetilde{u}_i^{(l)}(t_{i+1})$, we need to substitute $\widetilde{u}_i^{(l)}(t_{i+1})$ into Eqs. (21a)–(21c) to proceed the whole recurrence calculation. Obviously, we have $\widetilde{u}_i^{(l)}(t_{i+1}) = \widetilde{u}_{i+1}^{(l)}(t_{i+1})$.

After substituting $\widetilde{u}_i^{(l)}(t_i)$ into Eq. (17), we can obtain all needed numerical displacements (i.e., $u_i(t_i)$), velocities (i.e., $u_i^{(1)}(t_i)$), and accelerations (i.e., $u_i^{(2)}(t_i)$).

4 Implementation of cubic B-spline on MDOF systems

The dynamic equilibrium equations governing a linear MDOF system can be expressed as

$$\boldsymbol{M}\boldsymbol{U}^{(2)}(t) + \boldsymbol{D}\boldsymbol{U}^{(1)}(t) + \boldsymbol{K}\boldsymbol{U}(t) = \boldsymbol{F}(t), \tag{28}$$

where $\boldsymbol{M}$, $\boldsymbol{D}$, and $\boldsymbol{K}$ are the mass, damping, and stiffness matrices, respectively. $\boldsymbol{F}$ is the vector of externally applied loads. $\boldsymbol{U}(t)$, $\boldsymbol{U}^{(1)}(t)$, and $\boldsymbol{U}^{(2)}(t)$ are the unknown displacement, velocity, and acceleration function vectors of the discrete nodes on the structure.

The initial conditions are $\boldsymbol{U}(t_0) = \boldsymbol{a}_0$, $\boldsymbol{U}^{(1)}(t_0) = \boldsymbol{a}_1$, and $\boldsymbol{U}^{(2)}(t_0) = \boldsymbol{a}_2$.

As previously discussed in Section 3 for the SDOF system, Eq. (16) is used to represent the approximate solution to Eq. (28). First, we let

$$\boldsymbol{U}^{(l)}(t) = [\boldsymbol{u}_1^{(l)}(t) \quad \boldsymbol{u}_2^{(l)}(t) \quad \cdots \quad \boldsymbol{u}_j^{(l)}(t) \quad \cdots \quad \boldsymbol{u}_N^{(l)}(t)]^{\mathrm{T}}, \quad l = 0, 1, 2. \tag{29}$$

Assume that $u_{j,i}(t)$ are the cubic B-spline interpolation functions of the exact solution $u_j(t)$ within any time interval $I_i'\,(i=0,1,2,\cdots,n-1)$. Then, by use of Eq. (16), we have

$$u_{j,i}^{(l)}(t)=\sum_{k=-3}^{0}C_{j,k}^{i}B_{k,3}^{(l)}(\tau_i),\quad i=0,1,2,\cdots,n-1,\quad j=1,2,\cdots,N,\quad l=0,1,2. \tag{30}$$

Substituting Eq. (30) into Eq. (29) gives the approximate function vectors of the exact solution $\boldsymbol{U}^{(l)}(t)$ within any time interval I_i' as

$$\boldsymbol{U}_i^{(l)}(t)=\sum_{k=-3}^{0}B_{k,3}^{(l)}(\tau_i)\boldsymbol{C}_k^i,\quad i=0,1,2,\cdots,n-1,\quad l=0,1,2, \tag{31}$$

where $\boldsymbol{C}_k^i$ are the unknown coefficient vectors of size $N\times 1$. $\boldsymbol{C}_k^i$ can be written as

$$\boldsymbol{C}_k^i=[C_{1,k}^i\quad C_{2,k}^i\quad\cdots\quad C_{j,k}^i\quad\cdots\quad C_{N,k}^i]^{\mathrm{T}},\quad k=-3,-2,-1,0. \tag{32}$$

Similar to the SDOF system, here, we let

$$\widetilde{\boldsymbol{U}}_i^{(l)}(t)=(\varphi\Delta t)^{(l)}\cdot\boldsymbol{U}_i^{(l)}(t). \tag{33}$$

By substituting $t=t_i$ (i.e., $\tau_i=0$) and $t=t_{i+\varphi}$ (i.e., $\tau_i=1$) into Eq. (31) and using Eq. (33) and Table 1, we have

$$\widetilde{\boldsymbol{U}}_i(t_i)=\frac{1}{6}\boldsymbol{C}_{-3}^i+\frac{2}{3}\boldsymbol{C}_{-2}^i+\frac{1}{6}\boldsymbol{C}_{-1}^i, \tag{34a}$$

$$\widetilde{\boldsymbol{U}}_i^{(1)}(t_i)=-\frac{1}{2}\boldsymbol{C}_{-3}^i+0\boldsymbol{C}_{-2}^i+\frac{1}{2}\boldsymbol{C}_{-1}^i, \tag{34b}$$

$$\widetilde{\boldsymbol{U}}_i^{(2)}(t_i)=\boldsymbol{C}_{-3}^i-2\boldsymbol{C}_{-2}^i+\boldsymbol{C}_{-1}^i, \tag{34c}$$

and

$$\widetilde{\boldsymbol{U}}_i(t_{i+\varphi})=\frac{1}{6}\boldsymbol{C}_{-2}^i+\frac{2}{3}\boldsymbol{C}_{-1}^i+\frac{1}{6}\boldsymbol{C}_{0}^i, \tag{35a}$$

$$\widetilde{\boldsymbol{U}}_i^{(1)}(t_{i+\varphi})=-\frac{1}{2}\boldsymbol{C}_{-2}^i+0\boldsymbol{C}_{-1}^i+\frac{1}{2}\boldsymbol{C}_{0}^i, \tag{35b}$$

$$\widetilde{\boldsymbol{U}}_i^{(2)}(t_{i+\varphi})=\boldsymbol{C}_{-2}^i-2\boldsymbol{C}_{-1}^i+\boldsymbol{C}_{0}^i. \tag{35c}$$

With Eqs. (34a)–(34c), $\boldsymbol{C}_{-3}^i$, $\boldsymbol{C}_{-2}^i$, and $\boldsymbol{C}_{-1}^i$ can be calculated by

$$\boldsymbol{C}_{-3}^i=\widetilde{\boldsymbol{U}}_i(t_i)-\widetilde{\boldsymbol{U}}_i^{(1)}(t_i)+\frac{1}{3}\widetilde{\boldsymbol{U}}_i^{(2)}(t_i), \tag{36a}$$

$$\boldsymbol{C}_{-2}^i=\widetilde{\boldsymbol{U}}_i(t_i)+0\widetilde{\boldsymbol{U}}_i^{(1)}(t_i)-\frac{1}{6}\widetilde{\boldsymbol{U}}_i^{(2)}(t_i), \tag{36b}$$

$$\boldsymbol{C}_{-1}^i=\widetilde{\boldsymbol{U}}_i(t_i)+\widetilde{\boldsymbol{U}}_i^{(1)}(t_i)+\frac{1}{3}\widetilde{\boldsymbol{U}}_i^{(2)}(t_i). \tag{36c}$$

Prescribing that Eq. (28) is satisfied at $t = t_{i+\varphi}$ (i.e., $\tau_i = 1$) yields

$$\boldsymbol{M}\boldsymbol{U}_i^{(2)}(t_{i+\varphi}) + \boldsymbol{D}\boldsymbol{U}_i^{(1)}(t_{i+\varphi}) + \boldsymbol{K}\boldsymbol{U}_i(t_{i+\varphi}) = \boldsymbol{F}(t_{i+\varphi}), \quad i = 0, 1, 2, \cdots, n-1. \tag{37}$$

Using Eq. (33), Eq. (37) can be further expressed as

$$\boldsymbol{M}\widetilde{\boldsymbol{U}}_i^{(2)}(t_{i+\varphi}) + \varphi\Delta t \cdot \boldsymbol{D}\widetilde{\boldsymbol{U}}_i^{(1)}(t_{i+\varphi}) + (\varphi\Delta t)^2 \cdot \boldsymbol{K}\widetilde{\boldsymbol{U}}_i(t_{i+\varphi}) = (\varphi\Delta t)^2 \cdot \boldsymbol{F}(t_{i+\varphi}). \tag{38}$$

Substituting Eqs. (35a)–(35c) into Eq. (38) yields

$$\boldsymbol{\alpha}\boldsymbol{C}_{-2}^i + \boldsymbol{\beta}\boldsymbol{C}_{-1}^i + \boldsymbol{\gamma}\boldsymbol{C}_0^i = (\varphi\Delta t)^2 \cdot \boldsymbol{F}(t_{i+\varphi}), \quad i = 0, 1, 2, \cdots, n-1, \tag{39}$$

where

$$\boldsymbol{\alpha} = \frac{(\varphi\Delta t)^2}{6}\boldsymbol{K} - \frac{\varphi\Delta t}{2}\boldsymbol{D} + \boldsymbol{M}, \tag{40}$$

$$\boldsymbol{\beta} = \frac{2(\varphi\Delta t)^2}{3}\boldsymbol{K} - 2\boldsymbol{M}, \tag{41}$$

$$\boldsymbol{\gamma} = \frac{(\varphi\Delta t)^2}{6}\boldsymbol{K} + \frac{\varphi\Delta t}{2}\boldsymbol{D} + \boldsymbol{M}. \tag{42}$$

$\boldsymbol{C}_{-3}^i$, $\boldsymbol{C}_{-2}^i$, and $\boldsymbol{C}_{-1}^i$ have been previously solved by Eqs. (36a)–(36c). Thus, we can solve $\boldsymbol{C}_0^i$ by

$$\boldsymbol{C}_0^i = \boldsymbol{\gamma}^{-1}((\varphi\Delta t)^2 \cdot \boldsymbol{F}(t_{i+\varphi}) - \boldsymbol{\alpha}\boldsymbol{C}_{-2}^i - \boldsymbol{\beta}\boldsymbol{C}_{-1}^i), \quad i = 0, 1, 2, \cdots, n-1. \tag{43}$$

Then, substituting $t = t_{i+1}$ into Eqs. (31) and (33) yields

$$\widetilde{\boldsymbol{U}}_i^{(l)}(t_{i+1}) = \sum_{k=-3}^{0} (\varphi\Delta t)^{(l)} \cdot B_{k,3}^{(l)}\Big(\frac{1}{\varphi}\Big)\boldsymbol{C}_k^i, \quad i = 0, 1, 2, \cdots, n-1, \quad l = 0, 1, 2. \tag{44}$$

In Eq. (44), to complete the recurrence procedure, we let $\widetilde{\boldsymbol{U}}_{i+1}^{(l)}(t_{i+1}) = \widetilde{\boldsymbol{U}}_i^{(l)}(t_{i+1})$. Meanwhile, after substituting $\widetilde{\boldsymbol{U}}_{i+1}^{(l)}(t_{i+1})$ into Eq. (33), we can obtain $\widetilde{\boldsymbol{U}}_{i+1}^{(l)}(t_{i+1})(l = 0, 1, 2, i = 0, 1, 2, \cdots, n-1)$.

To clarify the complete algorithm of the proposed method, we give the calculation procedure of the MDOF system from the perspective of computer programming in Table 2.

Table 2 illustrates that all the $\boldsymbol{C}_k^i$ depend on the previous values already calculated. Therefore, the proposed algorithm is an explicit one. From Table 2, we can see that only one symmetric matrix inversion needs to be conducted within each time step. Moreover, compared with the methods of Refs. [9] and [15], all intermediate variables $\boldsymbol{C}_k^i$ have simpler forms and can be easily solved, which, to a great extent, can save computation time greatly. Thus, theoretically, the proposed method has the more desirable computation speed than the methods of Refs. [9] and [15].

5 Numerical stability and accuracy analyses

5.1 Stability analysis

In general, any global equation of motion can be degraded into a set of uncoupled SDOF systems by use of the modal decomposition. Meanwhile, the integration of the uncoupled equations is equivalent to the integration of the original system. Therefore, to study the stability properties of the proposed method, it suffices to only consider the SDOF system.

Table 2 Calculation process of proposed algorithm for dynamic response of MDOF systems

A Initial calculation

(i) Form the stiffness matrix $\boldsymbol{K}$, mass matrix $\boldsymbol{M}$, and damping matrix $\boldsymbol{D}$ of the system

(ii) Initialize $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$ as the displacement and velocity vectors. Then, ascertain the initial acceleration vector $\boldsymbol{a}_2$ using the relationship as follows:

$$\boldsymbol{a}_2 = \boldsymbol{M}^{-1}(\boldsymbol{F}(t_0) - \boldsymbol{D}\boldsymbol{a}_1 - \boldsymbol{K}\boldsymbol{a}_0)$$

(iii) Select the appropriate algorithmic parameter $\varphi\,(1 \leqslant \varphi < 2)$ and the time increment $(\Delta t < \Delta t_{\text{critical}})$. Then, specify $(\varphi\Delta t)^2 \cdot \boldsymbol{F}(t_{i+\varphi})\,(i = 0, 1, 2, \cdots, n-1)$

(iv) Determine constant matrices $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ as follows:

$$\boldsymbol{\alpha} = \frac{(\varphi\Delta t)^2}{6}\boldsymbol{K} - \frac{\varphi\Delta t}{2}\boldsymbol{D} + \boldsymbol{M}, \quad \boldsymbol{\beta} = \frac{2(\varphi\Delta t)^2}{3}\boldsymbol{K} - 2\boldsymbol{M}, \quad \boldsymbol{\gamma} = \frac{(\varphi\Delta t)^2}{6}\boldsymbol{K} + \frac{\varphi\Delta t}{2}\boldsymbol{D} + \boldsymbol{M}$$

(v) Calculate the inverse of symmetric matrix $\boldsymbol{\gamma}$, $\widetilde{\boldsymbol{\gamma}} = \boldsymbol{\gamma}^{-1}$

(vi) Calculate three unknown vectors of coefficients $\boldsymbol{C}_{-3}^0$, $\boldsymbol{C}_{-2}^0$, and $\boldsymbol{C}_{-1}^0$ using the terms as follows:

$$\boldsymbol{C}_{-3}^0 = \boldsymbol{a}_0 - \boldsymbol{a}_1 + \tfrac{1}{3}\boldsymbol{a}_2,$$
$$\boldsymbol{C}_{-2}^0 = \boldsymbol{a}_0 + 0\boldsymbol{a}_1 - \tfrac{1}{6}\boldsymbol{a}_2,$$
$$\boldsymbol{C}_{-1}^0 = \boldsymbol{a}_0 + \boldsymbol{a}_1 + \tfrac{1}{3}\boldsymbol{a}_2$$

(vii) Calculate $\boldsymbol{C}_0^0$ using the following relationship:

$$\boldsymbol{C}_0^0 = \widetilde{\boldsymbol{\gamma}}$$

B For each time step $(i = 0, 1, 2, \cdots, n-1)$

(i) Calculate the displacement vector and its two derivative vectors at $t = t_{i+1}$ by

$$\widetilde{\boldsymbol{U}}_i^{(l)}(t_{i+1}) = \sum_{k=-3}^{0} (\varphi\Delta t)^{(l)} \cdot B_{k,3}^{(l)}\left(\tfrac{1}{\varphi}\right)\boldsymbol{C}_k^i, \quad l = 0, 1, 2$$

(ii) Calculate three unknown coefficients $\boldsymbol{C}_{-3}^{i+1}$, $\boldsymbol{C}_{-2}^{i+1}$, and $\boldsymbol{C}_{-1}^{i+1}$ by

$$\boldsymbol{C}_{-3}^{i+1} = \widetilde{\boldsymbol{U}}_i(t_{i+1}) - \widetilde{\boldsymbol{U}}_i^{(1)}(t_{i+1}) + \tfrac{1}{3}\widetilde{\boldsymbol{U}}_i^{(2)}(t_{i+1}),$$
$$\boldsymbol{C}_{-2}^{i+1} = \widetilde{\boldsymbol{U}}_i(t_{i+1}) + 0\widetilde{\boldsymbol{U}}_i^{(1)}(t_{i+1}) - \tfrac{1}{6}\widetilde{\boldsymbol{U}}_i^{(2)}(t_{i+1}),$$
$$\boldsymbol{C}_{-1}^{i+1} = \widetilde{\boldsymbol{U}}_i(t_{i+1}) + \widetilde{\boldsymbol{U}}_i^{(1)}(t_{i+1}) + \tfrac{1}{3}\widetilde{\boldsymbol{U}}_i^{(2)}(t_{i+1})$$

(iii) Calculate $\boldsymbol{C}_0^{i+1}$ using the following relationship:

$$\boldsymbol{C}_0^{i+1} = \widetilde{\boldsymbol{\gamma}}((\varphi\Delta t)^2 \cdot \boldsymbol{F}(t_{i+1+\varphi}) - \boldsymbol{\alpha}\boldsymbol{C}_{-2}^{i+1} - \boldsymbol{\beta}\boldsymbol{C}_{-1}^{i+1})$$

(iv) Determine $\boldsymbol{U}_i(t_{i+1})$, $\boldsymbol{U}_i^{(1)}(t_{i+1})$, and $\boldsymbol{U}_i^{(2)}(t_{i+1})$ by

$$\boldsymbol{U}_i^{(m)}(t_{i+1}) = \widetilde{\boldsymbol{U}}_i^{(m)}(t_{i+1})/(\varphi\Delta t)^{(m)}, \quad m = 0, 1, 2$$

Note that analysis about the selection of algorithmic parameter φ is given in Section 5

The proposed method transfers the state at the ith step to the $(i+1)$th step as elucidated in Section 3. Clearly, the recurrence formula of the proposed algorithm for any SDOF system can be intrinsically written as

$$\widetilde{\boldsymbol{u}}_{i+1} = \boldsymbol{A}\widetilde{\boldsymbol{u}}_i + \widetilde{\boldsymbol{f}}_i, \tag{45}$$

where $\boldsymbol{A}$ is the amplification matrix, $\widetilde{\boldsymbol{u}}_{i+1}$ and $\widetilde{\boldsymbol{u}}_i$ are given by

$$\widetilde{\boldsymbol{u}}_i = [\widetilde{u}_i(t_i) \quad \widetilde{u}_i^{(1)}(t_i) \quad \widetilde{u}_i^{(2)}(t_i)]^{\mathrm{T}}, \tag{46}$$

$$\widetilde{\boldsymbol{u}}_{i+1} = [\widetilde{u}_i(t_{i+1}) \quad \widetilde{u}_i^{(1)}(t_{i+1}) \quad \widetilde{u}_i^{(2)}(t_{i+1})]^{\mathrm{T}}. \tag{47}$$

In addition, we need to define $\widetilde{\boldsymbol{u}}_{i+\varphi}$ as

$$\widetilde{\boldsymbol{u}}_{i+\varphi} = [\widetilde{u}_i(t_{i+\varphi}) \quad \widetilde{u}_i^{(1)}(t_{i+\varphi}) \quad \widetilde{u}_i^{(2)}(t_{i+\varphi})]^{\mathrm{T}}. \tag{48}$$

To find the amplification matrix $\boldsymbol{A}$, at first, we need to construct the relationships among $\widetilde{\boldsymbol{u}}_i$, $\widetilde{\boldsymbol{u}}_{i+1}$, and $\widetilde{\boldsymbol{u}}_{i+\varphi}$ by use of coefficients $C_k^i\,(k = -3, -2, -1, 0)$.

If we solve Eqs. (20a)–(20c) to reach C_{-2}^i, C_{-1}^i, and C_0^i, we would have

$$C_{-2}^i = \left[1 \quad -1 \quad \frac{1}{3}\right]\widetilde{\boldsymbol{u}}_{i+\varphi}, \tag{49a}$$

$$C_{-1}^i = \left[1 \quad 0 \quad -\frac{1}{6}\right]\widetilde{\boldsymbol{u}}_{i+\varphi}, \tag{49b}$$

$$C_0^i = \left[1 \quad 1 \quad \frac{1}{3}\right]\widetilde{\boldsymbol{u}}_{i+\varphi}. \tag{49c}$$

Similarly, with Eqs. (21a)–(21c), C^i_{-3}, C^i_{-2}, and C^i_{-1} can be represented by

$$C^i_{-3} = \left[1 \quad -1 \quad \frac{1}{3}\right]\widetilde{\boldsymbol{u}}_i, \tag{50a}$$

$$C^i_{-2} = \left[1 \quad 0 \quad -\frac{1}{6}\right]\widetilde{\boldsymbol{u}}_i, \tag{50b}$$

$$C^i_{-1} = \left[1 \quad 1 \quad \frac{1}{3}\right]\widetilde{\boldsymbol{u}}_i. \tag{50c}$$

Combining Eqs. (49a)–(49c) with Eqs. (50a)–(50c) gives

$$\begin{bmatrix} 1 & -1 & \dfrac{1}{3} \\ 1 & 0 & -\dfrac{1}{6} \end{bmatrix}\widetilde{\boldsymbol{u}}_{i+\varphi} = \begin{bmatrix} 1 & 0 & -\dfrac{1}{6} \\ 1 & 1 & \dfrac{1}{3} \end{bmatrix}\widetilde{\boldsymbol{u}}_i. \tag{51}$$

In addition, substituting Eq. (49c) and Eqs. (50b)–(50c) into Eq. (24) renders

$$\left[1 \quad 1 \quad \frac{1}{3}\right]\widetilde{\boldsymbol{u}}_{i+\varphi} + [\varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_3]\widetilde{\boldsymbol{u}}_i = \frac{(\varphi\Delta t)^2}{\gamma}\cdot f(t_{i+\varphi}), \tag{52}$$

where

$$\varepsilon_1 = \frac{\alpha+\beta}{\gamma}, \quad \varepsilon_2 = \frac{\beta}{\gamma}, \quad \varepsilon_3 = \frac{2\beta-\alpha}{\gamma}. \tag{53}$$

Actually, Eq. (52) is the equivalent form of Eq. (22). Equations (51) and (52) can be expressed in a unified form of

$$\widetilde{\boldsymbol{u}}_{i+\varphi} = \boldsymbol{A}_1\widetilde{\boldsymbol{u}}_i + \boldsymbol{f}_i, \tag{54}$$

where

$$\boldsymbol{A}_1 = \begin{bmatrix} \dfrac{5-\varepsilon_1}{6} & \dfrac{4-\varepsilon_2}{6} & \dfrac{7-6\varepsilon_3}{36} \\ -\dfrac{1+\varepsilon_1}{6} & -\dfrac{\varepsilon_2}{2} & \dfrac{1-6\varepsilon_3}{12} \\ -(1+\varepsilon_1) & -(\varepsilon_2+2) & -\dfrac{5+6\varepsilon_3}{6} \end{bmatrix}, \quad \boldsymbol{f}_i = \begin{bmatrix} \dfrac{(\varphi\Delta t)^2}{6\gamma}\cdot f(t_{i+\varphi}) \\ \dfrac{(\varphi\Delta t)^2}{2\gamma}\cdot f(t_{i+\varphi}) \\ \dfrac{(\varphi\Delta t)^2}{\gamma}\cdot f(t_{i+\varphi}) \end{bmatrix}. \tag{55}$$

With Eq. (27), $\widetilde{\boldsymbol{u}}_{i+1}$ can be written as

$$\widetilde{\boldsymbol{u}}_{i+1} = \widetilde{\boldsymbol{B}}_1\begin{bmatrix} C^i_{-3} \\ C^i_{-2} \\ C^i_{-1} \end{bmatrix} + \widetilde{\boldsymbol{B}}_2\begin{bmatrix} C^i_{-2} \\ C^i_{-1} \\ C^i_0 \end{bmatrix}, \tag{56}$$

where

$$\widetilde{\boldsymbol{B}}_1 = \begin{bmatrix} \widetilde{B}_{-3,3}(1/\varphi) & \widetilde{B}_{-2,3}(1/\varphi) & \widetilde{B}_{-1,3}(1/\varphi) \\ \widetilde{B}^{(1)}_{-3,3}(1/\varphi) & \widetilde{B}^{(1)}_{-2,3}(1/\varphi) & \widetilde{B}^{(1)}_{-1,3}(1/\varphi) \\ \widetilde{B}^{(2)}_{-3,3}(1/\varphi) & \widetilde{B}^{(2)}_{-2,3}(1/\varphi) & \widetilde{B}^{(2)}_{-1,3}(1/\varphi) \end{bmatrix}, \tag{57}$$

$$\widetilde{\boldsymbol{B}}_2 = \begin{bmatrix} 0 & 0 & \widetilde{B}_{0,3}(1/\varphi) \\ 0 & 0 & \widetilde{B}^{(1)}_{0,3}(1/\varphi) \\ 0 & 0 & \widetilde{B}^{(2)}_{0,3}(1/\varphi) \end{bmatrix}. \tag{58}$$

With Eqs. (49a)–(49c) and Eqs. (50a)–(50c), we have

$$[C_{-2}^i \quad C_{-1}^i \quad C_0^i]^{\mathrm{T}} = \boldsymbol{P}\widetilde{\boldsymbol{u}}_{i+\varphi}, \tag{59}$$

$$[C_{-3}^i \quad C_{-2}^i \quad C_{-1}^i]^{\mathrm{T}} = \boldsymbol{P}\widetilde{\boldsymbol{u}}_i, \tag{60}$$

where

$$\boldsymbol{P} = \begin{bmatrix} 1 & -1 & \dfrac{1}{3} \\ 1 & 0 & -\dfrac{1}{6} \\ 1 & 1 & \dfrac{1}{3} \end{bmatrix}. \tag{61}$$

Then, after substituting Eqs. (59) and (60) into Eq. (56) and using Eq. (54), we have

$$\begin{aligned} \widetilde{\boldsymbol{u}}_{i+1} &= \widetilde{\boldsymbol{B}}_1\boldsymbol{P}\widetilde{\boldsymbol{u}}_i + \widetilde{\boldsymbol{B}}_2\boldsymbol{P}\widetilde{\boldsymbol{u}}_{i+\varphi} = \widetilde{\boldsymbol{B}}_1\boldsymbol{P}\widetilde{\boldsymbol{u}}_i + \widetilde{\boldsymbol{B}}_2\boldsymbol{P}(\boldsymbol{A}_1\widetilde{\boldsymbol{u}}_i + \boldsymbol{f}_i) \\ &= (\widetilde{\boldsymbol{B}}_1\boldsymbol{P} + \widetilde{\boldsymbol{B}}_2\boldsymbol{P}\boldsymbol{A}_1)\widetilde{\boldsymbol{u}}_i + \widetilde{\boldsymbol{B}}_2\boldsymbol{P}\boldsymbol{f}_i = \boldsymbol{A}\widetilde{\boldsymbol{u}}_i + \widetilde{\boldsymbol{f}}_i. \end{aligned} \tag{62}$$

In Eq. (62), the amplification matrix $\boldsymbol{A}$ (i.e., $\widetilde{\boldsymbol{B}}_1\boldsymbol{P} + \widetilde{\boldsymbol{B}}_2\boldsymbol{P}\boldsymbol{A}_1$) is used to investigate stability of the proposed algorithm. Because the amplification matrix $\boldsymbol{A}$ contains too many terms, the concrete expression of $\boldsymbol{A}$ is not presented here. In fact, we can easily obtain $\boldsymbol{A}$ by virtue of some general mathematical softwares.

In this study, the stability analysis is conducted by solving the eigenvalue problem of the amplification matrix $\boldsymbol{A}$. The eigenvalues of $\boldsymbol{A}$ are obtained by solving $|\boldsymbol{A} - \lambda\boldsymbol{I}| = 0$, where $\boldsymbol{I}$ is the identity matrix of size 3×3. Here, let $\lambda_i\,(i = 1, 2, 3)$ be the eigenvalues of $\boldsymbol{A}$. To obtain a convergent algorithm, the norm of all eigenvalues should be less than unity, and thus we should have $\rho(\boldsymbol{A}) = \max(\|\lambda_1\|, \|\lambda_2\|, \|\lambda_3\|) \leqslant 1$, where $\rho(\boldsymbol{A})$ is called the spectral radius.

For the proposed step-by-step method, it suffices to investigate $\rho(\boldsymbol{A})$ only at $\xi = 0$. At this moment, $\rho(\boldsymbol{A})$ is a function only in terms of $\omega\Delta t$ and the algorithmic parameter φ. The natural frequency ω satisfies $\omega = 2\pi/T$, where T is the period of the considered system. To ensure that the algorithm in question is stable, the effects of the parameter φ on the spectral radius $\rho(\boldsymbol{A})$ need to be investigated. Here, we conduct this study through numerical experimentation instead of rigorous mathematical operation.

The variation of the spectral radius as a function of $\Delta t/T$ is shown in Fig. 3. For the convenience of comparison, only the spectral radius curves of some representative parameter values are shown in Fig. 3, where the spectral radius curves move downward with the increase in the parameter φ. Clearly, when $\varphi \geqslant 1.37$, the spectral radius achieves unconditional stability. However, increasing the value of φ would undoubtedly lower interpolation accuracy of the proposed method. Therefore, we give the suggested parameter condition as $1.0 \leqslant \varphi \leqslant 1.4$. Furthermore, we provide five parameter cases of the cubic B-spline method as

Case Ⅰ $\varphi = 1.00$, $\Delta t/T \leqslant 0.550$;
Case Ⅱ $\varphi = 1.20$, $\Delta t/T \leqslant 0.764$;
Case Ⅲ $\varphi = 1.30$, $\Delta t/T \leqslant 1.175$;
Case Ⅳ $\varphi = 1.37$ (unconditionally stable);
Case Ⅴ $\varphi = 1.40$ (unconditionally stable).

In Fig. 4, the radii from the cubic B-spline method are especially compared with the radii from the Noh-Bathe method[15]. Clearly, the cubic B-spline method is more flexible in controlling the stability and stable region than the Noh-Bathe method. In Fig. 4, for the Noh-Bathe method, when the algorithmic parameter p is equal to 0.5, the spectral radius remains to be 1 for a slightly wider range than the one from any other given case. However, compared with

the central difference method schemes, Case I has a much wider stable region. The proposed scheme gives very similar spectral radii as the Wilson-θ method, but possesses different calculation procedure which is formulated based on the uniform B-spline interpolation.

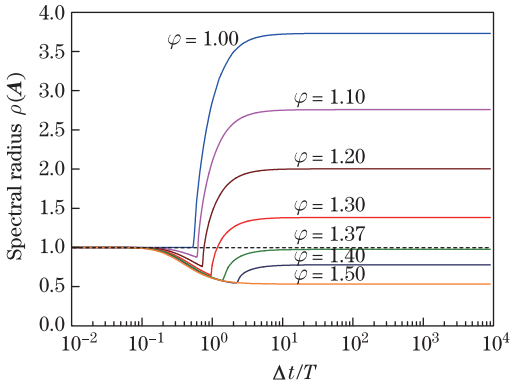


Fig. 3 Spectral radii of proposed method for various φ

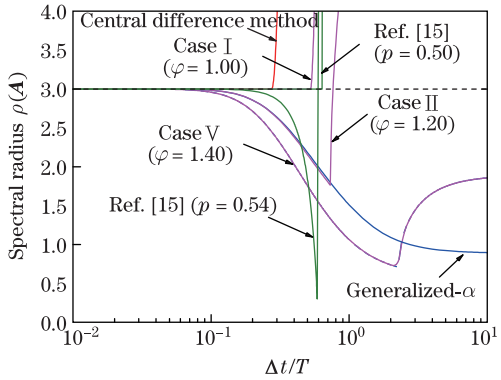
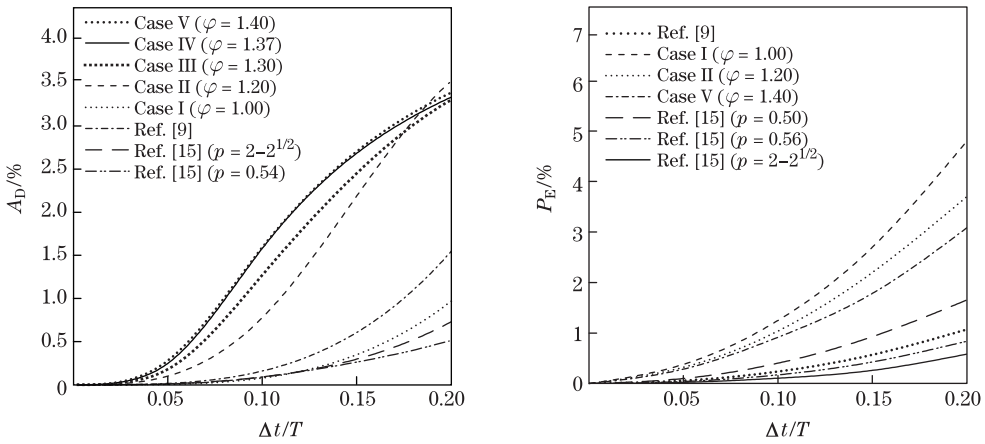


Fig. 4 Spectral radius comparison between proposed method and other schemes

5.2 Accuracy analysis

In general, a standard procedure used to estimate the numerical error of a step-by-step time integration method is to quantify the difference between numerical displacements and theoretical displacements for the free vibration of an undamped SDOF system. For the free vibration problem, there are two definitions for error estimation: amplitude decay (A_{D}) and period elongation (P_{E}). A_{D} is sometimes reported as the "algorithmic damping ratio".

In Fig. 5, the algorithmic damping ratio and period elongation curves from the cubic B-spline method are compared with the curves from the Noh-Bathe method[15] and the Bathe-Noh method[9]. Figure 5(a) illustrates that the algorithmic damping ratio from the cubic B-spline method increases with the increase in the parameter φ. Figure 5(b) shows that the period elongation from the cubic B-spline method decreases with the increase in the parameter φ. Clearly, the Noh-Bathe method of Ref. [15] produces the smaller damping ratio and period elongation than the proposed method. Figure 5(b) shows that the Bathe-Noh method of Ref. [9] produces the larger period elongation than the proposed method. However, considering that all curves in Fig. 5 are close, the proposed method has the desirable algorithmic damping ratio and period elongation.



(a) Algorithmic damping ratio (b) Period elongation

Fig. 5 Accuracy analysis

6 Numerical examples

To demonstrate validity of the proposed method for dynamic analysis, four numerical examples have been tested in this section, and five parameter cases previously given in Subsection 5.1 are used for computation. In addition, for comparison, the generalized-α ($\alpha_f = \frac{1}{3}, \alpha_m = 0, \beta = \frac{1}{2}, \gamma = \frac{5}{6}$) method[2], the Bathe-Noh method[9], and the Noh-Bathe method[15] are used in this section. Here, it is important to note that, for the Noh-Bathe method, we adopt the suggested algorithmic parameter value $p = 0.54$ for computation[15].

6.1 Standard undamped system

To test the calculation accuracy of the proposed scheme, the free vibration of a standard undamped SDOF system is considered. This system is

$$u^{(2)}(t) + \omega^2 u(t) = 0, \quad u(0) = 1, \quad u^{(1)}(0) = 0, \tag{63}$$

where $\omega = \pi$, that is, $T = 2$.

A time duration of $t = 2$ s is considered and calculated for accuracy and efficiency tests, and the global error norms E_l ($l = 0, 1, 2$) are employed and defined as

$$E_l = \sqrt{\sum_{i=0}^{n-1} (u_i^{(l)} - \widetilde{u}_i^{(l)})^2 \Big/ \sum_{i=0}^{n-1} (\widetilde{u}_i^{(l)})^2} \times 100\%, \tag{64}$$

in which $u_i^{(l)}$ are numerical results at the time t_i, and $\widetilde{u}_i^{(l)}$ are the corresponding exact ones.

In Fig. 6, global errors of various schemes are plotted in a log form. It can be noted that accuracy of the proposed scheme decreases with the increase in the parameter φ. Case V shows acceptable accuracy compared with other considered methods.

6.2 SDOF system

An SDOF system for numerical simulation is given by

$$u^{(2)}(t) + 4u^{(1)}(t) + 5u(t) = \sin(2t), \quad u(0) = 57/65, \quad u^{(1)}(0) = 2/65, \tag{65}$$

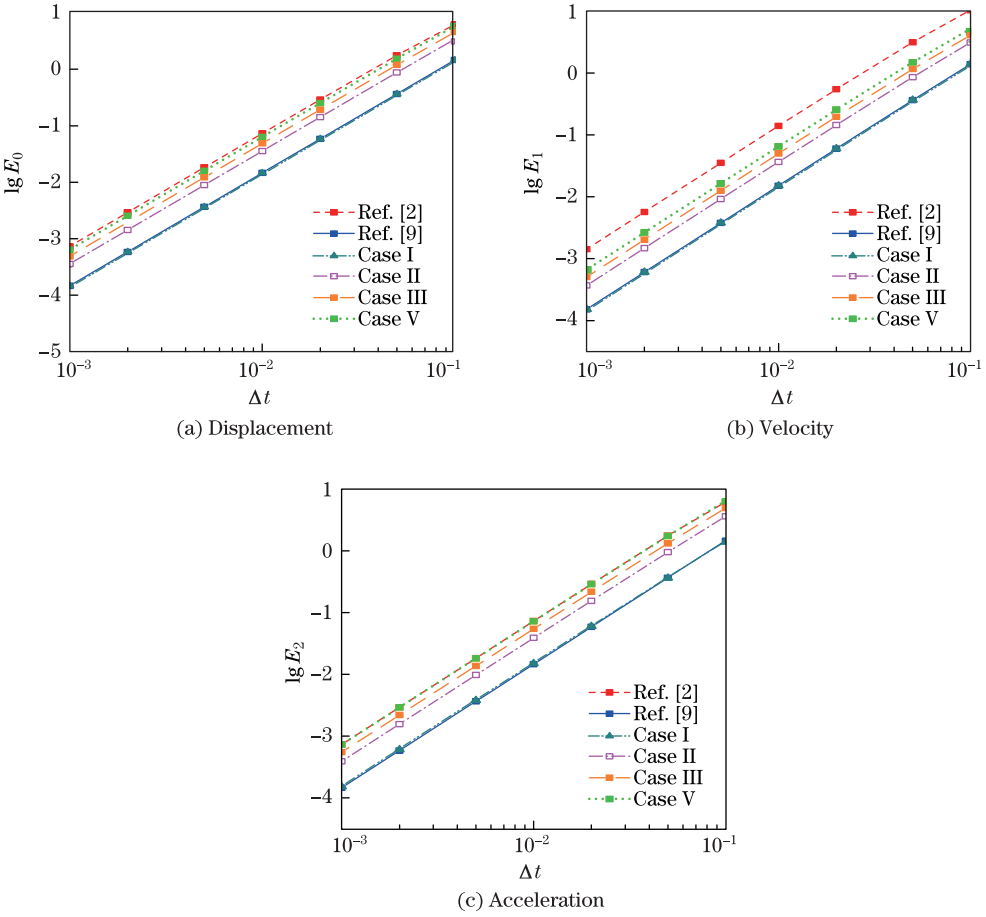
whose exact solution is $u(t) = \mathrm{e}^{-2t}(\cos t + 2\sin t) - (8\cos(2t) - \sin(2t))/65$.

To investigate the quantified effects of ratio $\Delta t/T$ on calculation accuracy, we select four different time increments Δt for calculation and accordingly list the global errors thereof in

Table 3. Here, the global errors are defined by

$$E_i = \sqrt{\sum_{k=1}^{n_g}(u_{\text{num}}^{(i)}(t_k) - u_{\text{exact}}^{(i)}(t_k))^2 \Big/ \sum_{k=1}^{n_g}(u_{\text{exact}}^{(i)}(t_k))^2} \times 100\%, \quad i = 0, 1, 2, \tag{66}$$

where $u_{\text{num}}^{(i)}$ and $u_{\text{exact}}^{(i)}$ are the numerical result and the exact solution at some given time, respectively. n_{g} is the number of time steps. n_{g} is calculated by $n_{\text{g}} = t_{\text{a}}/\Delta t$, where t_{a} is the selected time duration. Here, we select $t_{\text{a}} = 4$ s for computation.



(a) Displacement (b) Velocity

(c) Acceleration

Fig. 6 Global errors of various implicit methods for Subsection 6.1

Table 3 shows that the calculation accuracy of the proposed method decreases with the increase in the parameter φ. Clearly, Case V gives roughly the same magnitude of displacements and velocities as the generalized-α method[2]. Nevertheless, the acceleration errors from Case V are almost one order of magnitude less than the generalized-α method. All the global errors from Case I and Case II are much smaller than those from the generalized-α method.

In Table 3, compared with the Noh-Bathe method[15], Case I produces slightly larger displacement errors, but gives clearly smaller velocity and acceleration errors. Table 3 shows that the Bathe method[9] possesses higher accuracy than the cubic B-spline method. However, note that the Bathe method adopts two sub-steps in each time step, and the computation time consumed by these two methods is definitely larger than other conventional methods. Thus,

Table 3 Global errors of proposed method and other given methods for Subsection 6.2

Global error	Δt/s	Generalized-α[2]	Bathe-Noh[9]	Noh-Bathe[15]	Cubic B-spline method		
					Case Ⅰ	Case Ⅱ	Case Ⅳ
E_0/%	0.20	2.305	0.454	0.606 6	1.344	2.034	2.648
	0.10	0.598	0.114	0.187 4	0.345	0.537	0.718
	0.05	0.152	0.029	0.049 7	0.088	0.138	0.187
	0.02	0.025	0.004	0.008 2	0.014	0.023	0.031
E_1/%	0.20	5.022	1.742	2.905 6	2.325	3.767	5.080
	0.10	1.301	0.422	0.799 8	0.565	0.906	1.230
	0.05	0.331	0.104	0.206 5	0.140	0.224	0.306
	0.02	0.054	0.017	0.033 5	0.022	0.036	0.049
E_2/%	0.20	23.510	1.631	3.844 8	3.705	5.903	7.885
	0.10	13.810	0.471	1.117 7	1.061	1.691	2.287
	0.05	7.505	0.128	0.310 1	0.288	0.460	0.627
	0.02	3.159	0.022	0.057 3	0.049	0.078	0.107

to conduct the computation efficiency analysis, we give the time consumption results of various methods in Table 4, where $\Delta t = 0.05$ s and three different values of n_{g} are considered for the time consumption analysis. Table 4 shows that the proposed method spends almost the same computation time as the generalized-α method. However, for the Bathe-Noh method[9] and Noh-Bathe method[15], much more computation time than the proposed method is consumed.

Table 4 Time consumption analysis for Subsection 6.2

Time step n_{g}	Generalized-α[2]	Bathe-Noh[9]	Noh-Bathe[15]	Cubic B-spline method	
				Case Ⅰ	Case Ⅴ
5 000	11.988 6	24.505 0	35.324 3	11.921 4	11.884 8
10 000	24.344 8	47.361 4	68.430 7	23.548 0	23.686 9
20 000	47.246 8	93.211 2	135.190 6	45.770 3	46.569 6

In Table 5, to quantify the efficiency differences between the proposed method and two given novel methods, various Δt are adopted for different methods. In the table, we define the relative errors for numerical results in the given time as

$$\eta_i = \frac{|u_{\mathrm{num}}^{(i)} - u_{\mathrm{exact}}^{(i)}|}{|u_{\mathrm{exact}}^{(i)}|} \times 100\%, \quad i = 0, 1, 2. \tag{67}$$

Table 5 shows that the cubic B-spline method could give almost the same magnitude of accuracy when roughly the same computation time as the Noh-Bathe method[15] is consumed. Tables 3–5 demonstrate that, compared with the Bathe-Noh method[9] (unconditionally stable) and Noh-Bathe method[15] (conditionally stable), the computation efficiency of the proposed method is acceptable.

6.3 Two-dimensional Howe truss under impact loads

A Howe truss under four concentrated impact loads is shown in Fig. 7. Material properties for all elements are shown in the figure. For comparison, the generalized-α method is used for computation. The least period of this system (T_{min}) is equal to 0.008 2 s. Thus, we select $\Delta t = 2 \times 10^{-3}$ s ($\Delta t \leqslant 0.550 T_{\mathrm{min}}$) for Case Ⅰ, and $\Delta t = 8 \times 10^{-3}$ s ($\Delta t \leqslant 1.175 T_{\mathrm{min}}$) for Case Ⅲ.

Table 5 Computation efficiency of proposed method and other given methods for Subsection 6.2

	Time/s	Bathe-Noh[9] ($\Delta t = 0.06$ s)	Noh-Bathe[15] ($\Delta t = 0.12$ s)	Cubic B-spline method ($\Delta t = 0.04$ s)		
				Case Ⅰ	Case Ⅱ	Case Ⅴ
η_0/%	1.2	0.028 0	0.515 0	0.116 0	0.288 0	0.485 0
	6.0	0.042 0	0.736 0	0.009 0	0.024 0	0.043 0
	12.0	0.084 0	0.742 0	0.038 0	0.097 0	0.165 0
	120.0	0.009 0	0.705 0	0.128 0	0.318 0	0.534 0
η_1/%	1.2	0.178 0	1.360 0	0.123 0	0.296 0	0.487 0
	6.0	0.256 0	0.960 0	0.058 0	0.145 0	0.244 0
	12.0	0.150 0	0.963 0	0.019 0	0.048 0	0.079 0
	120.0	0.066 0	0.965 0	0.002 0	0.007 0	0.011 0
η_2/%	1.2	0.617 0	0.671 0	0.138 0	0.421 0	0.759 0
	6.0	0.031 0	1.475 0	0.044 0	0.029 0	0.021 0
	12.0	0.037 0	2.687 0	0.015 0	0.044 0	0.112 0
	120.0	0.131 0	4.221 0	0.182 0	0.372 0	0.587 0
E_0/%	120.0	0.072 0	0.617 0	0.053 0	0.131 0	0.166 0
E_1/%	120.0	0.081 0	0.982 0	0.073 0	0.180 0	0.167 0
E_2/%	120.0	0.116 0	1.401 0	0.161 0	0.252 0	0.231 0
Time cost/s	600.0	49.231 6	35.371 9	36.215 6	35.805 8	36.118 4

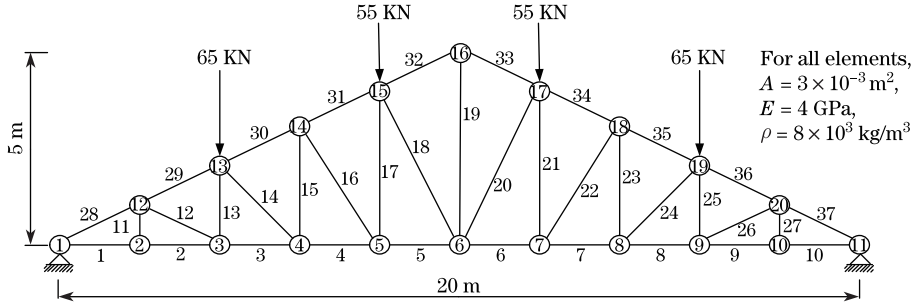


Fig. 7 Howe truss under impact loads

The time-history of vertical displacement over a time interval between 0 s and 2 s of Node 5 is illustrated in Fig. 8, and the horizontal displacement of Node 13 is plotted in Fig. 8. It is easily observed that the displacement curves from two conditional B-spline cases match very well with the curves from the generalized-α method[2]. As for Case Ⅴ, we can select a relatively large time increment Δt for computation as shown in Figs. 8 and 9, and thus the numerical results from Case Ⅰ are definitely less accurate than the other given B-spline cases. However, Figs. 8 and 9 show that all cases of cubic B-spline method are valid for dynamic responses.

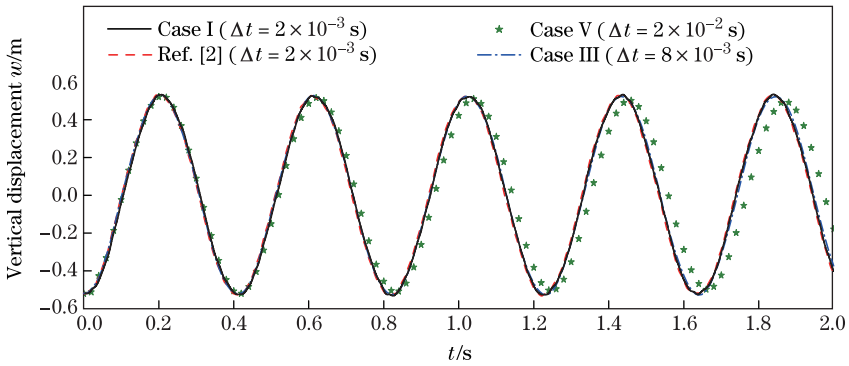


Fig. 8 Vertical displacement of Node 5

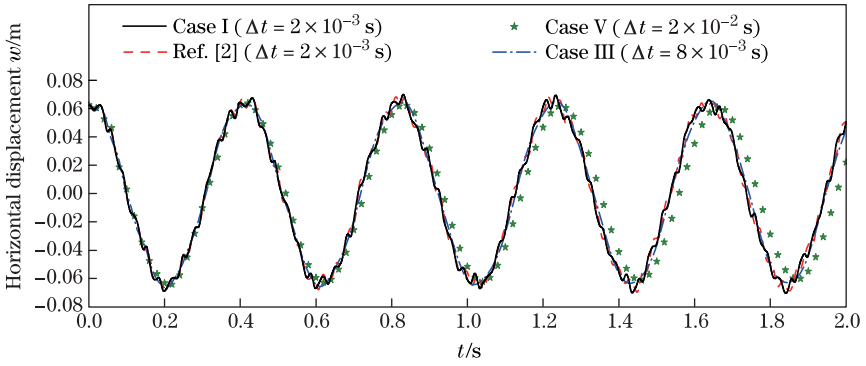


Fig. 9 Horizontal displacement of Node 13

As a sample, the results of time consumption for a complete analysis of Subsection 6.2 are tabulated in Table 6, where all methods use $\Delta t = 4\times10^{-4}$ s for computation. For different time steps n_{g}, the proposed method consumes almost the same time as the generalized-α method. By contrast, the Bathe-Noh method[9] and Noh-Bathe method[15] consume much more time than the proposed method. Furthermore, from Tables 4 and 6, we can safely conclude that the cubic B-spline method has high computation efficiency for the SDOF and MDOF systems.

Table 6 Time consumption analysis for Subsection 6.3

Time step n_{g}	Generalized-α[2]	Bathe-Noh[9]	Noh-Bathe[15]	Cubic B-spline method	
				Case I	Case V
5 000	16.863 3	30.403 8	46.378 4	16.596 1	16.637 4
10 000	66.280 7	123.153 0	187.833 4	65.866 9	66.955 3
20 000	168.704 3	313.252 0	477.770 6	166.390 3	166.945 7

6.4 Forced vibration of simply supported uniform continuous beam

To verify validity of the proposed method for the finite element method (FEM) calculation, a simply supported uniform continuous beam under a lateral concentrated variable load at the middle point is shown in Fig. 10. The dimensions and parameters used for analysis are the overall length $L = 8$ m, the radius of circular section $R = 2\times10^{-2}$ m, the cross-sectional area $A = \pi R^2$, the sectional inertia moment $I = \pi R^4/2$, Young's modulus $E = 100$ GPa, Poisson's ratio $\mu = 0.3$, the material density of the beam $\rho = 4\times10^{-4}$ kg/m^3, the damping ratio $\xi = 0$, and the lateral concentrated variable load $q(x,t) = F_0\sin(\omega_0 t)\cdot\delta(x - L/2)$. Here, we select $F_0 = 1$ kN, and $\omega_0 = 4$ rad/s. The analytical solution of the beam's bending deflection can be written as

$$w(x,t) = \frac{2F_0}{\rho AL}\sum_{r=1,3,5,\cdots}^{\infty}\frac{(-1)^{(r-1)/2}}{\omega_r^2(1-(\omega_0/\omega_r)^2)}\sin\frac{r\pi x}{L}\cdot\Big(\sin(\omega_0 t) - \frac{\omega_0}{\omega_r}\sin(\omega_r t)\Big), \tag{68}$$

where the natural frequency ω_r is determined by $\omega_r = \dfrac{r^2\pi^2}{L^2}\sqrt{\dfrac{EI}{\rho A}}$.

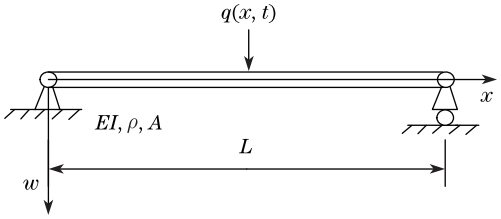


Fig. 10 Simply supported uniform continuous beam under lateral concentrated variable load

The equilibrium equation of this problem is formulated by use of the cubic Hermite finite element with its element number $N_s = 8$. The boundary conditions are satisfied by directly setting $w = 0$ at $x = 0$ and $x = L$. The least period of the equilibrium equation is equal to 0.005 6 s. Thus, similar to Subsection 6.2, we select $\Delta t = 3\times10^{-4}$ s for Case Ⅰ, $\Delta t = 5\times10^{-3}$ s for Case Ⅲ, and $\Delta t = 5\times10^{-2}$ s for Case Ⅳ. For simplicity, Fig. 11 just provides response values at the maximum deflection position (i.e., the middle point). In Fig. 11, although the various time increments Δt are used for different cubic B-spline cases, all curves from the proposed method are in desirable agreement with the exact curves.

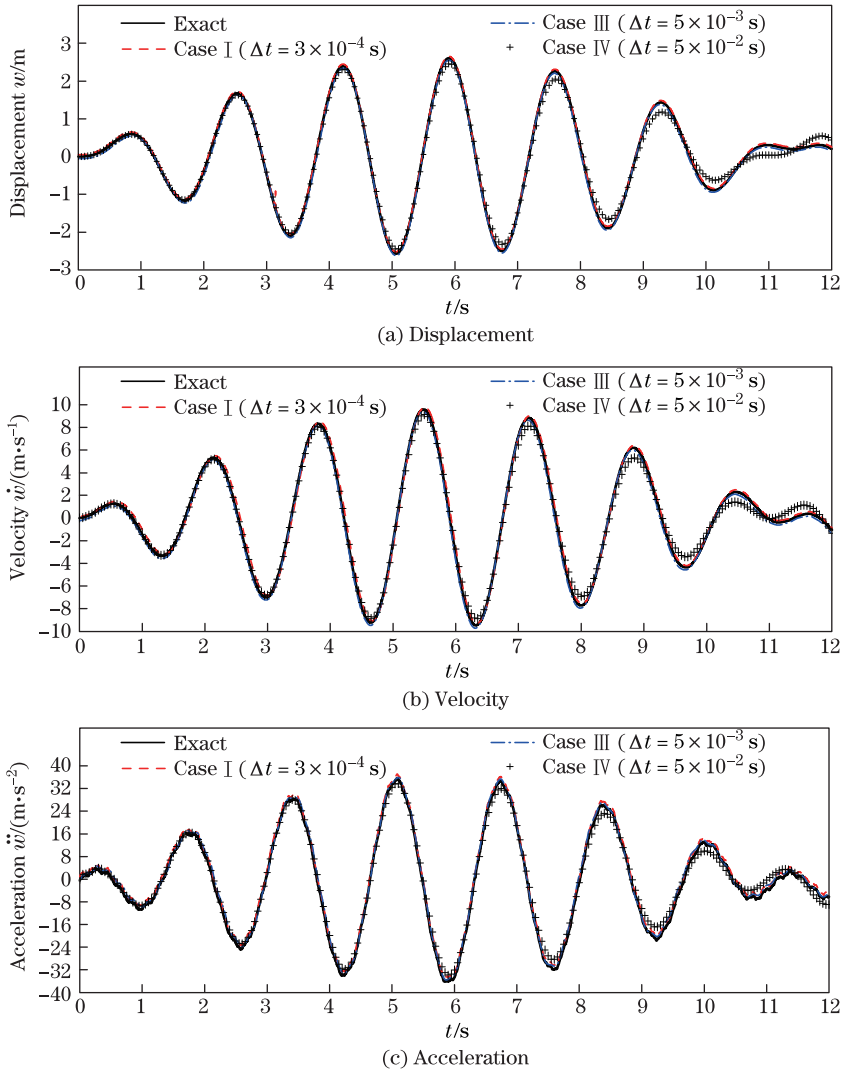


Fig. 11 Numerical results of different integration methods for Subsection 6.4

7 Conclusions

In this study, a new time integration method is proposed for the structural dynamic analysis by use of the cubic B-spline interpolation. By solving the differential equations of motion, a simple algorithm is formulated to calculate the dynamic responses of SDOF and MDOF systems.

The stability analysis shows that the proposed method can achieve both unconditional and conditional stability by adjusting the algorithmic parameter. The corresponding parameter conditions for conditional stability have been given in this study. The proposed method gives the acceptable numerical damping ratio and period elongation.

The validity of the proposed method has been confirmed with four numerical simulations. Time consumption results from numerical simulations demonstrate that, compared with other well-known methods, the proposed method has an appropriate computation speed. The computation efficiency analysis shows that the proposed method still possesses desirable computation efficiency compared with other schemes. Thus, the proposed method is a good choice for the structural dynamic analysis due to its high computation efficiency.

Actually, in terms of the capability of direct integration method for dealing with the nonlinear dynamic analysis, this method has potential ability to be generalized for a nonlinear system. This can be a good topic for the further research.

References

[1] Wilson, E. L., Farhoomand, I., and Bathe K. J. Nonlinear dynamic analysis of complex structures. *Earthquake Engineering and Structural Dynamics*, **1**, 241–252 (1972)

[2] Chung, J. and Hulbert G. M. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics*, **60**, 371–375 (1993)

[3] Bathe, K. J. Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Computers and Structures*, **85**, 437–445 (2007)

[4] Liu, T., Zhao, C., Li, Q., and Zhang, L. An efficient backward Euler time-integration method for nonlinear dynamic analysis of structures. *Computers and Structures*, **106/107**, 20–28 (2012)

[5] Dokainish, M. A. and Subbaraj, K. A survey of direct time-integration methods in computational structural dynamics I: explicit methods. *Computers and Structures*, **32**, 1371–1386 (1989)

[6] Mullen, R. and Belytschko, T. An analysis of an unconditionally stable explicit method. *Computers and Structures*, **16**, 691–696 (1983)

[7] Chung, J. and Lee, J. M. A new family of explicit time integration methods for linear and non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, **37**, 3961–3976 (1994)

[8] Subbaraj, K. and Dokainish, M. A. A survey of direct time-integration methods in computational structural dynamics II: implicit methods. *Computers and Structures*, **32**, 1387–1401 (1989)

[9] Bathe, K. J. and Noh, G. Insight into an implicit time integration scheme for structural dyanmics. *Computers and Structures*, **98/99**, 1–6 (2012)

[10] Rougier, E., Munjiza, A., and John, N. W. M. Numerical comparison of some explicit time integration schemes used in DEM, FEM/DEM and molecular dynamics. *International Journal for Numerical Methods in Engineering*, **61**, 856–879 (2004)

[11] Xie, Y. M. An assessment of time integration schemes for non-linear equations. *Journal of Sound and Vibration*, **192**, 321–331 (1996)

[12] Hulbert, G. M. and Chung, J. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Computer Methods in Applied Mechanics and Engineering*, **137**, 175–188 (1996)

[13] Chang, S. Y. and Liao, W. I. An unconditionally stable explicit method for structural dynamics. *Journal of Earthquake Engineering*, **9**, 349–370 (2005)

[14] Noh, G., Ham, S., and Bathe, K. J. Performance of an implicit time integration scheme in the analysis of wave propagations. *Computers and Structures*, **123**, 93–105 (2013)

[15] Noh, G. and Bathe, K. J. An explicit time integration scheme for the analysis of wave propagations. *Computers and Structures*, **129**, 178–193 (2013)

[16] Rostami, S., Shojaee, S., and Moeinadini, A. A parabolic acceleration time integration method for structural dynamics using quartic B-spline functions. *Applied Mathematical Modelling*, **36**, 5162–5182 (2012)

[17] Wen, W. B., Jian, K. L., and Luo, S. M. An explicit time integration method for structural dynamics using septuple B-spline functions. *International Journal for Numerical Methods in Engineering*, **97**, 629–657 (2014)

[18] Wen, W. B., Luo, S. M., and Jian, K. L. A novel time integration method for structural dynamics utilizing uniform quintic B-spline functions. *Archive of Applied Mechanics*, **85**, 1743–1759 (2015)

[19] Wen, W. B., Jian, K. L., and Luo, S. M. 2D numerical manifold method based on quartic uniform B-spline interpolation and its application in thin plate bending. *Applied Mathematics and Mechanics* (*English Edition*), **34**, 1017–1030 (2013) DOI 10.1007/s10483-013-1724-x