



# Development of a Direct Time Integration Method Based on Quartic B-Spline Collocation Method

Sobhan Rostami<sup>1</sup> · Saeed Shojaee<sup>2</sup>

Received: 6 April 2018 / Accepted: 22 October 2018 / Published online: 2 November 2018  
© Shiraz University 2018

## Abstract

A new version of quartic B-spline direct time integration method for dynamic analysis of structures is presented. This procedure is derived based on uniform quartic B-spline piecewise polynomial approximations and collocation method, named *Alpha Quartic B-spline* time integration method. In this way, at first, the method is implemented to solve the governing differential equation of motion of single-degree-of-freedom systems, and later, the proposed method is generalized for multi-degree-of-freedom systems. Stability and accuracy analysis of the proposed algorithm have been investigated completely. In the proposed algorithm by using two collocation parameters  $\alpha_1$  and  $\alpha_2$ , unconditional stability is achieved, but a local instability is created. The best values of these two parameters have been determined not only to maintain the stability, but also to ensure the desired accuracy. For accuracy analysis, dissipation and dispersion errors have been investigated for different cases of  $\alpha$ 's. Finally, for the proposed method, a simple step-by-step algorithm was presented. The effectiveness and robustness of the proposed algorithm in solving linear dynamic problems are demonstrated in the numerical examples.

**Keywords** Quartic B-spline · Direct time integration · Dynamic analysis · Stability · Dissipation · Dispersion

## 1 Introduction

Direct time integration algorithms are widely used in the computational analysis of structural dynamics and transient wave propagation problems. Generally, there are two basic categories of step-by-step integration methods. A time integration method is implicit if the solution procedure requires the factorization of an 'effective stiffness' matrix and is explicit otherwise (Dokainish and Subbaraj 1989; Subbaraj and Dokainish 1989). Both explicit and implicit methods have their own advantages and disadvantages.

Stability and accuracy are the most important characteristics of a time integration method. Depending on the stability characteristic, integration algorithms are classified as either unconditionally or conditionally stable, where unconditional stability implies that the numerical solution of a free vibration problem with any arbitrary initial conditions does not grow without bound for any integration time step size (Bathe 1996). In accuracy evaluation of the time integration methods, usually two quantities are determined: dispersion and dissipation. Dissipation (amplitude decay) and dispersion (period elongation) are two criteria used to evaluate the performance of an integration algorithm (Hilber and Hughes 1978).

Various different time integration algorithms have been proposed. The oldest and most powerful algorithms include the Newmark family of integration algorithms (Newmark 1959) and Wilson method (Wilson 1968; Wilson et al. 1972). Today, also new efficient methods have been developed. Recently, Bathe et al. proposed an efficient time integration method for solving the problems in structural dynamics and wave propagation. Desirable performance of the Bathe method was demonstrated in the series of

✉ Sobhan Rostami  
s-rostami@tvu.ac.ir

Saeed Shojaee  
saeed.shojaee@uk.ac.ir

<sup>1</sup> Department of Civil Engineering, Faculty of Shahid Chamran, Kerman Branch, Technical and Vocational University (TVU), Kerman, Iran

<sup>2</sup> Department of Civil Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

research papers (Bathe 2007; Bathe and Noh 2012; Noh and Bathe 2013).

Application of cubic and quartic B-spline for the numerical solution of dynamic systems has been presented by Rostami and Shojaee in a series of papers (Rostami and Shojaee 2017, 2018; Rostami et al. 2012, 2013; Shojaee et al. 2011, 2015). Implementation of cubic B-spline for the numerical solution of SDOF dynamic systems was introduced in Shojaee et al. (2011). Then in another work (Rostami et al. 2013), the proposed method was generalized for MDOF systems. Implementation of quartic B-spline for the numerical solution of dynamic systems, for the first time, has been done by Rostami et al. (2012). The proposed method has appropriate convergence, accuracy and low time consumption. The only shortcoming of this method was its conditional stability. Recently, in a new study (Shojaee et al. 2015), the proposed method was regenerated and developed to an unconditionally stable state; hence, it was named *modified quartic B-spline method*. Accuracy and stability analysis was profoundly performed in that paper. Numerical evaluations showed that the proposed method maintains its stability even in the analysis of structures with high frequencies. But some examples, in spite of having stable responses, do not show high accuracy, especially in the acceleration response of the couple systems. The present paper introduces a new version of the previous work and resolves this defect by the use of collocation scheme. Furthermore, in the new formulation, we eliminate the term derivation of acceleration, while we had to calculate it in each time step in the previous procedure (i.e., Rostami et al. 2012).

After our works in this field, Wen et al. (2017) proposed an explicit method by utilizing quartic B-splines. The Wen's proposed scheme possesses high calculation accuracy, but consumes more computation time compared to Bathe scheme. Malakiyeh et al. (2018) proposed an unconditionally stable method for linear analysis of structures based on Bezier curves and Bernstein polynomials. The important features of this method were low dissipation in the lower modes and high dissipation in the higher modes that leads to an increase in the accuracy of solutions.

## 2 Brief Review of the B-Splines

### 2.1 Definition of B-Splines

There are many definitions for B-splines. A detailed description of periodic B-spline functions can be found in Piegl and Tiller (2012) and Rogers (2001). For a simple

definition, let  $\Omega = \{x_0, x_1, x_2, \dots, x_n\}$  be a partition of  $[a, b] \subset \mathbb{R}$ . For the  $i$ th B-spline function of degree  $d$ , the basis function  $B_{i,d}(x)$  is defined by the Cox-de Boor recursion formulas. Specifically

$$B_{i,0}(x) = \begin{cases} 1, & x_i \leq x \leq x_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

and

$$B_{i,d}(x) = \left( \frac{x - x_i}{x_{d+i} - x_i} \right) B_{i,d-1}(x) + \left( \frac{x_{d+i+1} - x}{x_{d+i+1} - x_{i+1}} \right) B_{i+1,d-1}(x) \quad (2)$$

In this formula,  $B_{i,d}(x)$  is a polynomial of degree  $d$  on each interval  $x_i \leq x \leq x_{i+1}$  as  $B_{i,d}(x)$  and its derivatives of order 1, 2, ...,  $d - 1$  are all continuous over the entire domain. The values of  $x_i$  are elements of knot vector  $\Omega$  that must be monotonically increasing series of real numbers, i.e.,  $x_i \leq x_{i+1}$ . In the recursive formula,  $0/0 = 0$  is adopted (De Boor et al. 1978).

Furthermore, in the periodic type, each basis function is a simple transfer of the other one and the range of nonzero function values (support) spreads when the degree increases. Thus, the basis function provides support on the interval  $x_i \leq x \leq x_{i+d+1}$ . Figure 1 shows the B-spline functions. As depicted in Fig. 1, for a quartic B-spline used in this study, the usable parameter range of  $B_{i,4}$  is  $x_i \leq x \leq x_{i+5}$ .

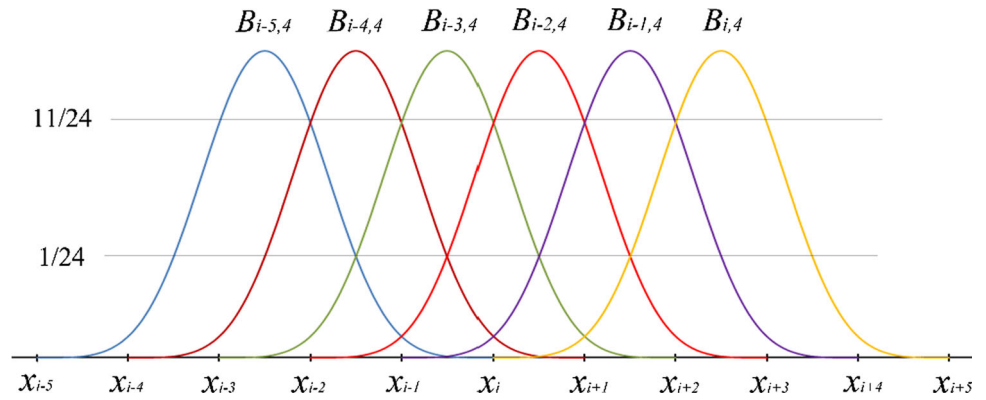
The B-spline of degree  $d$ ,  $B_{i,d}(x)$ , where  $i \in \mathbb{Z}$  possess the properties such as nonnegativity, i.e.,  $B_{i,d}(x) \geq 0 \forall x \in \mathbb{R}$ , and partition of unity, i.e.,  $\sum B_{i,d}(t) = 1 \forall [a_1, a_2] \in \mathbb{R}$ . However, for quartic periodic basis function we used in study, on each interval  $x_j \leq x \leq x_k$ , the range in which partition of unity properties is satisfied is  $a_1 = x_{j+4} \leq x \leq x_{k-4} = a_2$ . This means that in Fig. 1, with six basis functions considered, the usable range is  $a_1 = x_{i-1} \leq x \leq x_{i+1} = a_2$ .

### 2.2 Quartic B-Spline Interpolation

In this study, the uniform quartic B-spline with equidistantly distributed set of nodes  $\Omega$  over the domain  $[a, b]$ , and the subinterval length is defined by  $h = x_{i+1} - x_i$  ( $i = 0, 1, \dots, n - 1$ ). As discussed in Rostami et al. (2012), to construct the quartic B-splines, it is required to form the set of nodal points as  $x_{-4} < \dots < x_0 = a < \dots < x_n = b < \dots < x_{n+4}$ . For any node  $x_j$  ( $j = -4, -3, \dots, n + 4$ ), we have  $x_j = x_i + (j - i)h$ , and thus,  $B_{j,4}(x)$  is the shifted instance of  $B_{i,4}(x)$ , i.e.,  $B_{j,4}(x) = B_{i,4}(x - (j - i)h)$ .

Here, we apply Eq. (2) to get  $B_{i,4}(x)$ , ( $i = 0, 1, \dots, n - 1$ ) which are defined as

Fig. 1 Quartic B-splines



$$B_{i,4}(x) = \frac{1}{4!h^4} \begin{cases} (x-x_i)^4 & x_i \leq x \leq x_{i+1} \\ (x-x_i)^4 - 5(x-x_{i+1})^4 & x_{i+1} \leq x \leq x_{i+2} \\ (x-x_i)^4 - 5(x-x_{i+1})^4 + 10(x-x_{i+2})^4 & x_{i+2} \leq x \leq x_{i+3} \\ (x_{i+5}-x)^4 - 5(x_{i+4}-x)^4 & x_{i+3} \leq x \leq x_{i+4} \\ (x_{i+5}-x)^4 & x_{i+4} \leq x \leq x_{i+5} \end{cases} \quad (3)$$

Indeed, for any subinterval  $I_i \equiv [x_i, x_{i+1}]$ , to satisfy partition of unity properties of the basis functions, the usable piecewise quartic B-splines, as shown in Fig. 1, are  $B_{i-4,4}, \dots, B_{i,4}$ .

As we use these basis functions in a time-dependent problem, here the knot vector  $\Omega$  consists of time instants. Therefore,  $x$  in  $I$  represents  $t$ , i.e., time instants. Considering the fact that we will deal with them in the following section, it is necessary to have six basis functions in each time interval. Therefore, we consider each time interval  $I_i \equiv [t_i, t_{i+1}]$  twofold of the usual range to satisfy this issue. So for any time subinterval  $I_i$ , the usable piecewise quartic B-splines, as illustrated in Fig. 1, are  $B_{i-5,4}, B_{i-3,4}, \dots, B_{i,4}$ .

For ease, let  $\tau_i = (t - t_i)/\Delta t$  (i.e.,  $t = t_i + \tau_i \Delta t$ ), based upon the same assumption of Eq. (3), all usable uniform B-splines within any subinterval,  $I_i(\tau \in [t_i, t_{i+1}], \tau_i \in [0, 1])$ , as displayed in Fig. 2, can be rewritten as  $B_{-4,4}(\tau_i), B_{-3,4}(\tau_i), \dots, B_{1,4}(\tau_i)$ . Those B-splines are

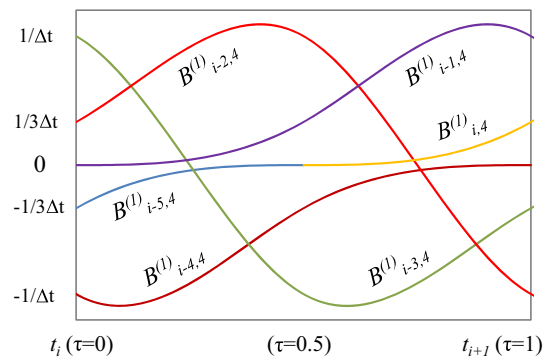


Fig. 3 First derivative of quartic B-spline

coupled with their derivatives with respect to variable  $t$ . Figures 3 and 4 show the quartic B-splines first and second derivatives, respectively. For  $0 \leq \tau \leq 0.5$ , these B-splines coupled with their  $n$ th derivatives with respect to variable  $t$  can be given as

$$B_{-4}^{(n)}(\tau_i) = \frac{(-\Delta t/2)^{-n}}{(4-n)!} (1-2\tau_i)^{4-n} \quad (4)$$

$$B_{-3}^{(n)}(\tau_i) = \frac{(-\Delta t/2)^{-n}}{(4-n)!} \left( (1-2\tau_i)^{4-n} - 5(1-2\tau_i)^{4-n} \right) \quad (5)$$

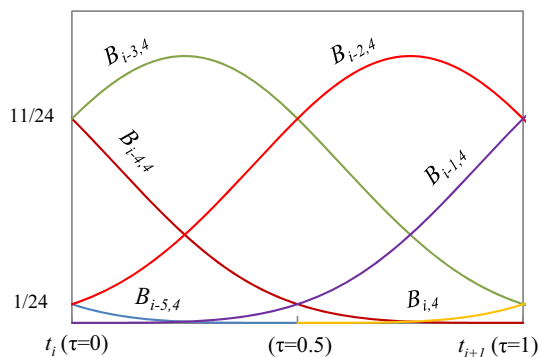


Fig. 2 Quartic B-spline

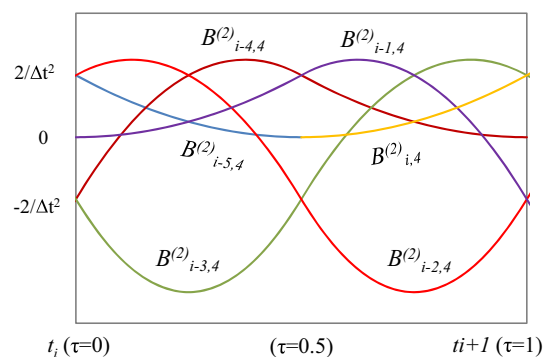


Fig. 4 Second derivative of quartic B-spline

$$B_{-2}^{(n)}(\tau_i) = \frac{(-\Delta t/2)^{-n}}{(4-n)!} \left( (3-2\tau_i)^{4-n} - 5(2-2\tau_i)^{4-n} + 10(1-2\tau_i)^{4-n} \right) \tag{6}$$

$$B_{-1}^{(n)}(\tau_i) = \frac{(\Delta t/2)^{-n}}{(4-n)!} \left( (1+2\tau_i)^{4-n} - 5(2\tau_i)^{4-n} \right) \tag{7}$$

$$B_0^{(n)}(\tau_i) = \frac{(\Delta t/2)^{-n}}{(4-n)!} (2\tau_i)^{4-n} \tag{8}$$

$$B_1^{(n)}(\tau_i) = 0 \tag{9}$$

And for  $0.5 \leq \tau \leq 1$  can be given as

$$B_{-4}^{(n)}(\tau_i) = 0 \tag{10}$$

$$B_{-3}^{(n)}(\tau_i) = \frac{(-\Delta t/2)^{-n}}{(4-n)!} (2-2\tau_i)^{4-n} \tag{11}$$

$$B_{-2}^{(n)}(\tau_i) = \frac{(-\Delta t/2)^{-n}}{(4-n)!} \left( (3-2\tau_i)^{4-n} - 5(2-2\tau_i)^{4-n} \right) \tag{12}$$

$$B_{-1}^{(n)}(\tau_i) = \frac{(-\Delta t/2)^{-n}}{(4-n)!} \left( (4-2\tau_i)^{4-n} - 5(3-2\tau_i)^{4-n} + 10(2-2\tau_i)^{4-n} \right) \tag{13}$$

$$B_0^{(n)}(\tau_i) = \frac{(\Delta t/2)^{-n}}{(4-n)!} \left( (2\tau_i)^{4-n} - 5(2\tau_i-1)^{4-n} \right) \tag{14}$$

$$B_1^{(n)}(\tau_i) = \frac{(\Delta t/2)^{-n}}{(4-n)!} (2\tau_i-1)^{4-n} \tag{15}$$

The quartic B-spline interpolation function for any interpolated interval  $I_i$  as shown in Fig. 2 is a linear combination of the quartic B-spline as follows:

$$S_i^{(n)}(t) = \sum_{j=-4}^1 C_{i+j} B_{i+j,4}^{(n)}(t) \tag{16}$$

where  $C_{i+j}$  (control points) are unknown real coefficients and  $B_{i,4}^{(n)}(t)$  is the  $n$ th derivative of fourth-degree (quartic) B-spline functions. This is three times differentiable and compatible. A detailed description of periodic B-spline can be found in Rogers (2001).

Having Eqs. (4–15) in hand, Eq. (16) can be extended to a matrix form as

$$S_i^{(n)}(t) = \mathbf{B}^{(n)}(\tau_i) \mathbf{C}_i, \quad t \in [t_i, t_{i+1}] \quad \tau_i \in [0, 1] \tag{17}$$

where

$$\mathbf{B}^{(n)}(\tau_i) = \left[ B_{-4,4}^{(n)}(\tau_i) \quad B_{-3,4}^{(n)}(\tau_i) \quad \cdots \quad B_{1,4}^{(n)}(\tau_i) \right] \tag{18}$$

$$\mathbf{C}_i = [C_{i-4} \quad C_{i-3} \quad \cdots \quad C_{i+1}]^T \tag{19}$$

**Table 1** Values of  $B_{i,4}, B_{i,4}^{(1)}, B_{i,4}^{(2)}$

	$t = t_i(\tau_i = 0)$			$t = t_{i+1}(\tau_i = 1)$		
	$n = 0$	$n = 1$	$n = 2$	$n = 0$	$n = 1$	$n = 2$
$B_{-4,4}^{(n)}$	1/24	-1/(3Δt)	2/Δt <sup>2</sup>	0	0	0
$B_{-3,4}^{(n)}$	11/24	-1/Δt	-2/Δt <sup>2</sup>	0	0	0
$B_{-2,4}^{(n)}$	11/24	1/Δt	-2/Δt <sup>2</sup>	1/24	-1/(3Δt)	2/Δt <sup>2</sup>
$B_{-1,4}^{(n)}$	1/24	1/(3Δt)	2/Δt <sup>2</sup>	11/24	-1/Δt	-2/Δt <sup>2</sup>
$B_{0,4}^{(n)}$	0	0	0	11/24	1/Δt	-2/Δt <sup>2</sup>
$B_{1,4}^{(n)}$	0	0	0	1/24	1/(3Δt)	2/Δt <sup>2</sup>

In the next sections, the quartic B-spline is used to construct a numerical solution of a second-order initial value problem. Thus, the evaluation of  $B_{i,4}^{(n)}$  ( $i = -4, -3, \dots, 1; n = 0, 1, 2$ ) at the discrete time instant  $t = t_i$  and  $t_{i+1}$  (i.e.,  $\tau_i = 0$  and 1) is needed. These values are listed in Table 1.

### 3 Implementation of Quartic B-Spline on SDOF Systems

A linear differential equation of motion can be expressed as  $u^{(2)}(t) + 2\xi\omega u^{(1)}(t) + \omega^2 u(t) = f(t)/m$  (20)

where  $m, \xi$  and  $\omega$  are the mass, damping ratio and natural frequency of the system.  $\ddot{u}(t), \dot{u}(t)$  and  $u(t)$  are the displacement, velocity and acceleration, respectively. This equation is subjected to the initial conditions:  $u(t_0) = u_0$  and  $\dot{u}(t_0) = v_0$ . The applied force,  $f(t)$ , is given by a set of discrete values  $f_i = f(t_i)$ .

According to Eq. (17), the approximate solution of Eq. (20), within any time interval  $\Delta t = t_{i+1} - t_i$ , can be expressed as

$$u^{(n)}(t) = \mathbf{B}^{(n)}(\tau_i) \mathbf{C}_i, \quad t \in [t_i, t_{i+1}] \quad \tau_i \in [0, 1] \tag{21}$$

According to Eq. (21), all discrete unknown physical variables  $u^{(n)}(t)$  can be degraded into the solving of  $\mathbf{C}_i$ . In fact, to determine Eq. (21), unknown coefficient vector  $\mathbf{C}_i$  needs to be solved. To avoid direct solution of  $\mathbf{C}_i$ , numerical displacement, velocity and acceleration at start and end points of the time interval  $[t_i, t_{i+1}]$  will be determined to represent  $\mathbf{C}_i$ . So, for  $n = 0, 1, 2$ , when substituting  $t = t_i$  (i.e.,  $\tau_i = 0$ ) and  $t = t_{i+1}$  (i.e.,  $\tau_{i+1} = 1$ ), we have

$$\bar{u}_i = \mathbf{J} \mathbf{C}_i \quad \text{or} \quad \mathbf{C}_i = \mathbf{J}^{-1} \bar{u}_i \tag{22}$$

where

$$J = \frac{1}{24} \begin{bmatrix} 1 & 11 & 11 & 1 & 0 & 0 \\ -8 & -24 & 24 & 8 & 0 & 0 \\ 48 & -48 & -48 & 48 & 0 & 0 \\ 0 & 0 & 1 & 11 & 11 & 1 \\ 0 & 0 & -8 & -24 & 24 & 8 \\ 0 & 0 & 48 & -48 & -48 & 48 \end{bmatrix} \quad (23)$$

$$\bar{u}_i = [\hat{u}_i^T \hat{u}_{i+1}^T]^T \quad (24a)$$

$$\hat{u}_i = [u(t_i) \quad \Delta t.u^{(1)}(t_i) \quad \Delta t^2.u^{(2)}(t_i)]^T \quad (24b)$$

$$\hat{u}_{i+1} = [u(t_{i+1}) \quad \Delta t.u^{(1)}(t_{i+1}) \quad \Delta t^2.u^{(2)}(t_{i+1})]^T \quad (24c)$$

Substituting Eq. (22) into Eq. (21) yields:

$$u^{(n)}(t) = B^{(n)}(\tau_i)J^{-1}\bar{u}_i, \quad n = 0, 1, 2 \quad (25)$$

Substituting Eq. (25) into Eq. (20), the equation of motion evaluated at time  $t_i$  can be expressed as

$$\begin{aligned} & (B^{(2)}(\tau_i) + 2\xi\omega B^{(1)}(\tau_i) + \omega^2 B^{(0)}(\tau_i))J^{-1}\bar{u}_i \\ & = f(t_i + \tau_i\Delta t)/m; \\ & t \in [t_i, t_{i+1}] \quad \tau_i \in [0, 1] \end{aligned} \quad (26)$$

In the above equation, we let

$$\begin{aligned} p(\tau_i) &= (B^{(2)}(\tau_i) + 2\xi\omega B^{(1)}(\tau_i) + \omega^2 B^{(0)}(\tau_i))J^{-1} \\ &= [p_1(\tau_i) \quad p_2(\tau_i) \quad \cdots \quad p_6(\tau_i)] \end{aligned} \quad (27)$$

Inserting Eq. (27) in Eq. (26) and using Eqs. (24a, 24b, 24c) can extract the following residual equation:

$$\begin{aligned} R(\tau_i) &= [p_1(\tau_i) \quad p_2(\tau_i) \quad p_3(\tau_i)]\hat{u}_i \\ &+ [p_4(\tau_i) \quad p_5(\tau_i) \quad p_6(\tau_i)]\hat{u}_{i+1} - f(t_i + \tau_i\Delta t)/m; \\ &t \in [t_i, t_{i+1}] \quad \tau_i \in [0, 1] \end{aligned} \quad (28)$$

Based on the collocation method, the residual Eq. (28) should satisfy  $R(\tau_i) = 0$  at  $\tau_i = 1$  (i.e.,  $t = t_{i+1}$ ),  $\tau_i = \alpha_1$  (i.e.,  $t = t_i + \alpha_1\Delta t$ ) and  $\tau_i = \alpha_2$  (i.e.,  $t = t_i + \alpha_2\Delta t$ ). So we have

$$\begin{aligned} [p_1(1)p_2(1)p_3(1)]\hat{u}_i + [p_4(1)p_5(1)p_6(1)]\hat{u}_{i+1} \\ = f(t_i + \Delta t)/m \end{aligned} \quad (29)$$

$$\begin{aligned} [p_1(\alpha_1)p_2(\alpha_1)p_3(\alpha_1)]\hat{u}_i + [p_4(\alpha_1)p_5(\alpha_1)p_6(\alpha_1)]\hat{u}_{i+1} \\ = f(t_i + \alpha_1\Delta t)/m \end{aligned} \quad (30)$$

$$\begin{aligned} [p_1(\alpha_2)p_2(\alpha_2)p_3(\alpha_2)]\hat{u}_i + [p_4(\alpha_2)p_5(\alpha_2)p_6(\alpha_2)]\hat{u}_{i+1} \\ = f(t_i + \alpha_2\Delta t)/m \end{aligned} \quad (31)$$

Equations (29–31) can be written in a matrix form as

$$\begin{aligned} & \begin{bmatrix} p_1(1) & p_2(1) & p_3(1) \\ p_1(\alpha_1) & p_2(\alpha_1) & p_3(\alpha_1) \\ p_1(\alpha_2) & p_2(\alpha_2) & p_3(\alpha_2) \end{bmatrix} \hat{u}_i \\ & + \begin{bmatrix} p_4(1) & p_5(1) & p_6(1) \\ p_4(\alpha_1) & p_5(\alpha_1) & p_6(\alpha_1) \\ p_4(\alpha_2) & p_5(\alpha_2) & p_6(\alpha_2) \end{bmatrix} \hat{u}_{i+1} \\ & = \frac{1}{m} \begin{bmatrix} f(t_i + \Delta t) \\ f(t_i + \alpha_1\Delta t) \\ f(t_i + \alpha_2\Delta t) \end{bmatrix} \end{aligned} \quad (32)$$

With matrix calculation, Eq. (32) will be transformed into a simple equation as

$$\hat{u}_{i+1} = A\hat{u}_i + \hat{f}_i \quad (33)$$

where

$$A = - \begin{bmatrix} p_4(1) & p_5(1) & p_6(1) \\ p_4(\alpha_1) & p_5(\alpha_1) & p_6(\alpha_1) \\ p_4(\alpha_2) & p_5(\alpha_2) & p_6(\alpha_2) \end{bmatrix}^{-1} \begin{bmatrix} p_1(1) & p_2(1) & p_3(1) \\ p_1(\alpha_1) & p_2(\alpha_1) & p_3(\alpha_1) \\ p_1(\alpha_2) & p_2(\alpha_2) & p_3(\alpha_2) \end{bmatrix} \quad (34)$$

$$\hat{f}_i = \begin{bmatrix} p_4(1) & p_5(1) & p_6(1) \\ p_4(\alpha_1) & p_5(\alpha_1) & p_6(\alpha_1) \\ p_4(\alpha_2) & p_5(\alpha_2) & p_6(\alpha_2) \end{bmatrix}^{-1} \cdot \frac{1}{m} \begin{bmatrix} f(t_i + \Delta t) \\ f(t_i + \alpha_1\Delta t) \\ f(t_i + \alpha_2\Delta t) \end{bmatrix} \quad (35)$$

In the above equation, the collocation parameters  $\alpha_1$  and  $\alpha_2$  have an important role in the stability and accuracy of the proposed method. In order that these matrices be invertible,  $\alpha_1$  and  $\alpha_2$  must satisfy the conditions  $0 < \alpha_1, \alpha_1 < 1$  and  $\alpha_1 \neq \alpha_2$ .

### 4 Implementation of Quartic B-Spline on MDOF Systems

The equilibrium equations governing the linear dynamic response of a system with multi-degrees of freedom are written as

$$MU^{(2)}(t) + DU^{(1)}(t) + KU(t) = F(t) \quad (36)$$

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

The initial conditions at  $t = 0$  are  $U(t_0) = U_0$ ,  $U^{(1)}(t_0) = \dot{U}_0$  and  $U^{(2)}(t_0) = \ddot{U}_0$ . So we have

$$\begin{aligned} U^{(n)}(t) &= [u_1^{(n)}(t) \quad u_2^{(n)}(t) \quad \cdots \quad u_k^{(n)}(t) \quad \cdots \quad u_N^{(n)}(t)]^T, \\ &n = 0, 1, 2 \end{aligned} \quad (37)$$

As it was discussed in the previous section for SDOF systems, here Eq. (21) can be employed to represent the approximate solution of Eq. (36). Hence, we have

$$u^{(n)}(t) = \mathbf{B}^{(n)}(\tau_i) \mathbf{C}_{i,k} \tag{38}$$

$$\mathbf{U}^{(n)}(t) = \hat{\mathbf{B}}^{(n)}(\tau_i) \hat{\mathbf{C}}_i \tag{39}$$

where

$$\hat{\mathbf{B}}^{(n)}(\tau_i) = \begin{bmatrix} \mathbf{B}^{(n)}(\tau_i) & & & \\ & \ddots & & \\ & & \mathbf{B}^{(n)}(\tau_i) & \\ & & & \mathbf{B}^{(n)}(\tau_i) \end{bmatrix}_{N \times 6N} \tag{40}$$

$$\hat{\mathbf{C}}_i = \begin{bmatrix} \mathbf{C}_{i,1}^T & \mathbf{C}_{i,2}^T & \cdots & \mathbf{C}_{i,N}^T \end{bmatrix}_{1 \times 6N}^T \tag{41}$$

Just here the unknown real coefficient  $\mathbf{C}_{i,k}$ s has an extra index  $k$  that introduces each degree of freedom, so that  $k = 1, 2, 3, \dots, N$ , where  $N$  is the number of all degrees of freedom. So  $\mathbf{C}_{i,k}$  is the unknown coefficients vector of size  $6 \times 1$ , and  $\hat{\mathbf{B}}^{(n)}(\tau_i)$  are block diagonal matrices with element  $\mathbf{B}^{(n)}(\tau_i)$ . Similar work was done in the previous section, by setting  $t = t_i$  (i.e.,  $\tau_i = 0$ ) and  $t = t_{i+1}$  (i.e.,  $\tau_{i+1} = 1$ ) in Eq. (39), we obtain

$$\bar{\mathbf{U}}_i = \bar{\mathbf{J}} \hat{\mathbf{C}}_i \quad \text{or} \quad \hat{\mathbf{C}}_i = \bar{\mathbf{J}}^{-1} \bar{\mathbf{U}}_i \tag{42}$$

where

$$\bar{\mathbf{U}}_i = [\hat{\mathbf{U}}_i^T \quad \hat{\mathbf{U}}_{i+1}^T]^T \tag{43a}$$

$$\hat{\mathbf{U}}_i = [\mathbf{U}(t_i) \quad \Delta t \mathbf{U}^{(1)}(t_i) \quad \Delta t^2 \mathbf{U}^{(2)}(t_i)]^T \tag{43b}$$

$$\hat{\mathbf{U}}_{i+1} = [\mathbf{U}(t_{i+1}) \quad \Delta t \mathbf{U}^{(1)}(t_{i+1}) \quad \Delta t^2 \mathbf{U}^{(2)}(t_{i+1})]^T \tag{43c}$$

$$\bar{\mathbf{J}}(\tau_i) = \begin{bmatrix} \begin{pmatrix} \hat{\mathbf{B}}(0) \\ \Delta t \hat{\mathbf{B}}^{(1)}(0) \\ \Delta t^2 \hat{\mathbf{B}}^{(2)}(0) \end{pmatrix} \\ \begin{pmatrix} \hat{\mathbf{B}}(1) \\ \Delta t \hat{\mathbf{B}}^{(1)}(1) \\ \Delta t^2 \hat{\mathbf{B}}^{(2)}(1) \end{pmatrix} \end{bmatrix}_{6N \times 6N} \tag{44}$$

In fact,  $\bar{\mathbf{J}}$  is a sparse and block matrix. It can be proved that, by the use of  $\mathbf{J}$  in Eq. (23), the calculation of  $\bar{\mathbf{J}}$  can be simplified as

$$\mathbf{J}^{-1} = [\mathbf{J}_1 \quad \mathbf{J}_2 \quad \mathbf{J}_3 \quad \mathbf{J}_4 \quad \mathbf{J}_5 \quad \mathbf{J}_6] \tag{45}$$

where  $\mathbf{J}_j$ 's ( $j = 1, 2, \dots, 6$ ), are  $1 \times 6$  column vectors of  $\mathbf{J}^{-1}$ . So

$$\bar{\mathbf{J}}^{-1} = \begin{bmatrix} \mathbf{J}_1 & & & & & \\ & \mathbf{J}_1 & & & & \\ & & \mathbf{J}_1 & & & \\ & & & \mathbf{J}_1 & & \\ & & & & \mathbf{J}_1 & \\ & & & & & \mathbf{J}_1 \end{bmatrix}_{6N \times 6N} \tag{46}$$

Inserting Eq. (41) into Eq. (39), we will have

$$\mathbf{U}^{(n)}(t) = \hat{\mathbf{B}}^{(n)}(\tau_i) \bar{\mathbf{J}}^{-1} \bar{\mathbf{U}}_i \tag{47}$$

Substituting Eq. (47) into equilibrium equations, i.e., Eq. (36), gives the following residual equation:

$$\mathbf{R}(\tau_i) = (\mathbf{M} \hat{\mathbf{B}}^{(2)}(\tau_i) + \mathbf{D} \hat{\mathbf{B}}^{(1)}(\tau_i) + \mathbf{K} \hat{\mathbf{B}}(\tau_i)) \bar{\mathbf{J}}^{-1} \bar{\mathbf{U}}_i - \mathbf{F}(t) \tag{48}$$

By setting

$$\begin{aligned} & (\mathbf{M} \hat{\mathbf{B}}^{(2)}(\tau_i) + \mathbf{D} \hat{\mathbf{B}}^{(1)}(\tau_i) + \mathbf{K} \hat{\mathbf{B}}(\tau_i)) \bar{\mathbf{J}}^{-1} \\ & = [\mathbf{p}_1(\tau_i) \quad \mathbf{p}_2(\tau_i) \quad \cdots \quad \mathbf{p}_6(\tau_i)] \end{aligned} \tag{49}$$

$\mathbf{p}_k(\tau_i)$  ( $k = 1, 2, \dots, 6$ ) are all symmetric matrices of size  $N \times N$ . Equation (48) can be written as

$$\begin{aligned} \mathbf{R}(\tau_i) &= [\mathbf{p}_1(\tau_i) \quad \mathbf{p}_2(\tau_i) \quad \mathbf{p}_6(\tau_i)] \hat{\mathbf{U}}_i \\ &+ [\mathbf{p}_1(\tau_i) \quad \mathbf{p}_2(\tau_i) \quad \mathbf{p}_6(\tau_i)] \hat{\mathbf{U}}_{i+1} - \mathbf{F}(t) \end{aligned} \tag{50}$$

As discussed and illustrated in the previous section, after acting the collocation method on Eq. (48), it is possible to extract the recurrence formula as

$$\hat{\mathbf{U}}_{i+1} = -\mathbf{A}_1^{-1} \mathbf{A}_2 \hat{\mathbf{U}}_i + \mathbf{A}_1^{-1} \hat{\mathbf{f}}_i = \mathbf{A} \hat{\mathbf{U}}_i + \hat{\mathbf{F}}_i \tag{51}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{p}_4(1) & \mathbf{p}_5(1) & \mathbf{p}_6(1) \\ \mathbf{p}_4(\alpha_1) & \mathbf{p}_5(\alpha_1) & \mathbf{p}_6(\alpha_1) \\ \mathbf{p}_4(\alpha_2) & \mathbf{p}_5(\alpha_2) & \mathbf{p}_6(\alpha_2) \end{bmatrix}, \tag{52}$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{p}_1(1) & \mathbf{p}_2(1) & \mathbf{p}_3(1) \\ \mathbf{p}_1(\alpha_1) & \mathbf{p}_2(\alpha_1) & \mathbf{p}_3(\alpha_1) \\ \mathbf{p}_1(\alpha_2) & \mathbf{p}_2(\alpha_2) & \mathbf{p}_3(\alpha_2) \end{bmatrix}$$

$$\hat{\mathbf{f}}_i = \begin{bmatrix} \mathbf{F}(t_i + \Delta t) \\ \mathbf{F}(t_i + \alpha_1 \Delta t) \\ \mathbf{F}(t_i + \alpha_2 \Delta t) \end{bmatrix} \tag{53}$$

Although in linear problems the inversion of  $\mathbf{A}_1$  is done only once and can be efficiently solved by using some general mathematical software, for a system with large degrees of freedom direct inversion of  $\mathbf{A}_1$  is not desirable. To solve this shortcoming, a technique which has been proposed in Wen et al. (2015) is used here to pass the direct inversion of matrix  $\mathbf{A}_1$ . At first, a well-known theorem and corollary is introduced (Lu and Shiou 2002).

**Theorem** Let a non-singular square matrix  $\tilde{\mathbf{Q}}$  be partitioned into  $2 \times 2$  blocks as

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{bmatrix} \tag{54}$$

Consider  $\mathbf{Q}_4$  non-singular; then, the matrix  $\tilde{\mathbf{Q}}$  is invertible if and only if the Schur complement  $\mathbf{Q}_1 - \mathbf{Q}_2 \mathbf{Q}_4^{-1} \mathbf{Q}_3$  is invertible; meanwhile, we have



$$\tilde{Q}^{-1} = \begin{bmatrix} Q_5^{-1} & -Q_5^{-1}Q_2Q_4^{-1} \\ -Q_4^{-1}Q_3Q_5^{-1} & Q_4^{-1} + Q_4^{-1}Q_3Q_5^{-1}Q_2Q_4^{-1} \end{bmatrix} \quad (55)$$

$$Q_5 = Q_1 - Q_2Q_4^{-1}Q_3 \quad (56)$$

**Corollary** Consider  $Q_4$  non-singular; then,  $Q_5$  is invertible if matrix  $\tilde{Q}$  is invertible.

With the above theorem and corollary, the invertible matrix  $A_1$  can be expressed as

$$A_1 = \begin{bmatrix} S_1 & S_2 \\ S_3 & S_4 \end{bmatrix}; \quad S_4 = \begin{bmatrix} p_5(\alpha_1) & p_6(\alpha_1) \\ p_5(\alpha_2) & p_6(\alpha_2) \end{bmatrix} \quad (57)$$

According to the form considered for  $S_4$  as defined in Eq. (57),

$$S_1 = p_4(1); \quad S_2 = [p_5(1) \quad p_6(1)]; \quad S_3 = \begin{bmatrix} p_4(\alpha_1) \\ p_4(\alpha_2) \end{bmatrix} \quad (58)$$

Then we have

$$A_1^{-1} = \begin{bmatrix} S_5^{-1} & -S_5^{-1}S_2S_4^{-1} \\ -S_4^{-1}S_3S_5^{-1} & S_4^{-1} + S_4^{-1}S_3S_5^{-1}S_2S_4^{-1} \end{bmatrix} \quad (59)$$

$$S_5 = S_1 - S_2S_4^{-1}S_3 \quad (60)$$

$S_4^{-1}$  can be directly solved by using Eq. (55). Thus, the solving of  $A_1^{-1}$  can be degraded into the inversion of three  $N \times N$  matrices. The proposed matrix inverse scheme is more efficient than conventional Gauss algorithm where more time is needed for inversion of asymmetric matrices.

In order to write a computer code, the complete algorithm used in this proposed method is given in Table 2.

## 5 Stability and Accuracy Analysis of the Proposed Method

In a time integration procedure such as the proposed method, independent parameters such as  $\alpha_1$  and  $\alpha_2$  not only control the stability, but also determine the accuracy. By investigating the properties of a numerical method such as

stability and accuracy, the most favorable values of these parameters can be obtained so that high accuracy is achieved, while unconditional stability is maintained.

### 5.1 Stability Analysis

In general, any global equation of motion can be degraded into a set of uncoupled SDOF systems by the 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 (Bathe 1996).

Stability analysis of an integration algorithm applied to linear elastic systems is generally carried out by the amplification matrix approach. An amplification matrix is formed for an integration algorithm, and the algorithm is considered to be unconditionally stable if the spectral radius of the amplification matrix does not exceed the value of 1.0 for any value of  $\omega_n \Delta t$ , where  $\omega_n$  is the highest natural frequency of the structure and  $\Delta t$  is the time step size used in the integration algorithm (Bathe 1996). Otherwise, the integration algorithm is considered to be conditionally stable. To this end, the equations of the proposed algorithms for a SDOF system under free vibration as elucidated in Eq. (33) can be represented by the following matrix recursive relationship:

$$\begin{Bmatrix} u(t_{i+1}) \\ \Delta t \cdot u^{(1)}(t_{i+1}) \\ \Delta t^2 \cdot u^{(2)}(t_{i+1}) \end{Bmatrix} = [A] \begin{Bmatrix} u(t_i) \\ \Delta t \cdot u^{(1)}(t_i) \\ \Delta t^2 \cdot u^{(2)}(t_i) \end{Bmatrix} + \hat{f}_i \quad (61)$$

where  $A$  is the amplification matrix which transfers the state at the time instant  $t_i$  to the state at the time instant  $t_{i+1}$ . Because  $A$  contains too many terms, the complete expression of  $A$  is not presented here. Actually, we can easily derive the expression of  $A$  by virtue of some general mathematical software.

It can be shown mathematically that the response produced by the recursive relation in Eq. (33) for any arbitrary

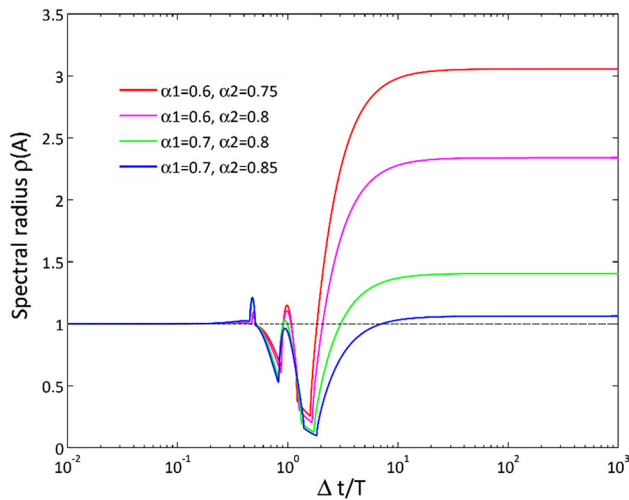
**Table 2** Step-by-step algorithm for calculation of response of MDOF systems using the proposed method

#### A. Initial calculation

1. Form stiffness matrix  $K$ , mass matrix  $M$  and damping matrix  $D$  of the system. Then calculate  $J^{-1}$  and  $\bar{J}^{-1}$  using Eq. (23) and (46)
2. Initialize  $U(t_0)$  and  $U^{(1)}(t_0)$  as displacement and velocity vectors. Then, determine the initial acceleration vector  $U^{(2)}(t_0)$  using Eq. (36)
3. Select an appropriate time step  $\Delta t$  and desirable parameters,  $\alpha_1$  and  $\alpha_2$ . Then determine  $\hat{U}_0$  using Eq. (43b)
4. Educe and compute constant matrices  $A$  employing Eq. (51) and the corresponding matrix calculation technique

#### B. For each time step

1. Specify the vector of applied forces  $\hat{F}_i$  to the system in each time instant using Eqs. (53) and (51)
2. Calculate  $\hat{U}_{i+1}$  using Eq. (51), and then, determine  $U(t_{i+1})$ ,  $U^{(1)}(t_{i+1})$  and  $U^{(2)}(t_{i+1})$  using Eq. (43c)



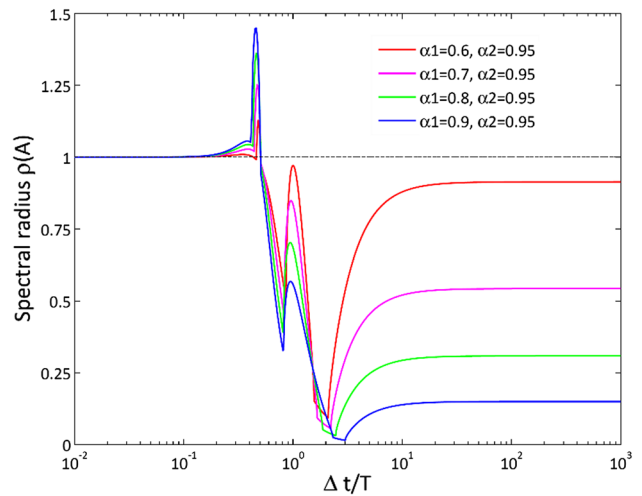
**Fig. 5** Variation in spectral radius in terms of variation in  $\Delta t/T$  (conditionally stable when  $\Delta t/T \rightarrow \infty$ )

initial conditions and  $\Delta t$  will be bounded if the magnitudes of all eigenvalues of the amplification matrix in  $\mathbf{A}$  are less than or equal to unity. In other words, the numerical scheme is stable if the spectral radius of  $A$ ,  $\rho(A) = \max(\lambda^i)$  where  $\lambda^i$  are the eigenvalues, is strictly less than unity.  $\rho(A)$  is the spectral radius which is a function of time step length,  $\Delta t$ .

Since the changes in damping ratio do not have much impact on the variation in spectral radius, here, assume  $\xi = 0$  in amplification matrix. So at this moment,  $\rho(A)$  is a function in terms of parameter  $\alpha_1$ ,  $\alpha_2$  and  $\omega\Delta t$ . To determine the stability of the proposed method, the effects of parameters  $\alpha_1$  and  $\alpha_2$  on the spectral radius  $\rho(A)$  need to be investigated.

Here the stability analysis is conducted through numerical experimentation, and some representative parameter combinations are discussed. Figures 5 and 6 depict the variation in spectral radius  $\rho(A)$  of amplification matrix of the proposed method in terms of variation in dimensionless time step  $\Delta t/T$  for different values of  $\alpha_1$ ,  $\alpha_2$ . As it is depicted in Figs. 5 and 6, for  $\Delta t < 0.2T$ , the proposed method is always stable. So the spectral radius curves remain 1.0 for different parameter combinations. Figures 5 and 6 demonstrate that the increase in  $\alpha_1$ ,  $\alpha_2$  converts conditional stability to unconditionally stable case.

Figure 5 and 6 show that there is an instability gap in the curves. As it is depicted in Fig. 6, range of  $0.24T < \Delta t < 0.5T$  is almost an instability interval. These figures show increasing the values of  $\alpha_1$  and  $\alpha_2$ , and decrease the value of spectral radius  $\rho(A)$  in the limit  $\Delta t/T \rightarrow \infty$ . Notably, the minimum points of these curves move downward and rightward as two algorithmic parameters increase, and the abrupt turning points in the



**Fig. 6** Variation in spectral radius in terms of variation in  $\Delta t/T$  (unconditionally stable when  $\Delta t/T \rightarrow \infty$ )

figure indicate that the eigenvalues of amplification matrix  $\mathbf{A}$  alter between real values and complex conjugate values. However, these small turnings have no substantial influence on algorithmic stability and high-frequency dissipation characteristics.

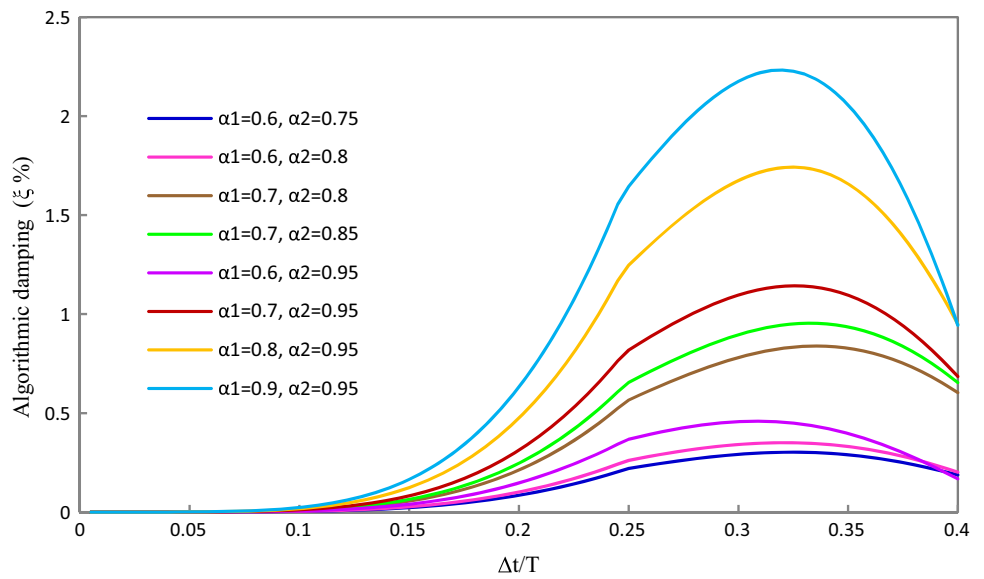
### 5.2 Accuracy Analysis

In order to show the accuracy of the proposed method, we need to evaluate the dissipation and dispersion error of the new scheme. So, we should evaluate the amplitude decay (AD) and period elongation (PE) for a periodical dynamic problem. As measures of the numerical dissipation and dispersion, we consider the *algorithmic damping ratio*  $\bar{\xi}$  and *relative period error*  $\tau = (\bar{T} - T)/T$ , respectively. Note that both  $\bar{\xi}$  and  $\bar{T}$  are defined in terms of the principal roots of characteristic equation derived from  $|A - \lambda I| = 0$ . Thus, these measures of accuracy are defined only about values  $\Omega$  so that stability region is  $0 < \Omega \leq \Omega_{crit}$ . Outside this region accuracy is not an issue and we are concerned only about stability.  $T = 2\pi/\omega$  and  $\bar{T} = 2\pi/\bar{\omega}$  are convenient measures of frequency distortion (dispersion) numerically introduced by the algorithm.

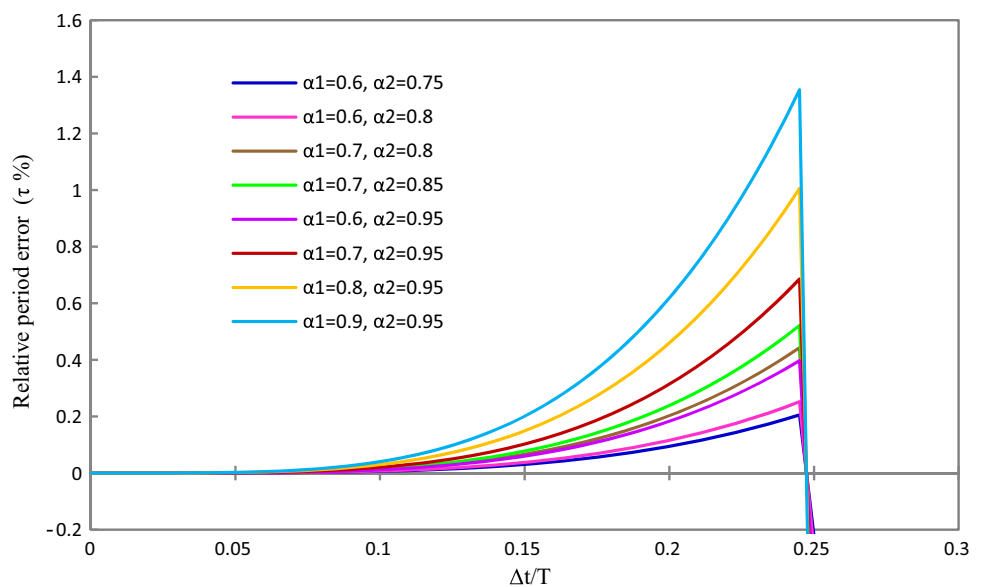
Figures 7 and 8 plot the curves of the relative error of period and the error related to the numerical damping or, in other words, the rate of amplitude decay versus the ratio of dimensionless time step  $\Delta t/T$  for different values of  $\alpha_s$ , respectively. The curves in Fig. 8 show that algorithmic damping of the proposed method for  $\Delta t = 0.33T$  has a peak value. Therefore, after that, the error decreases again. Also, in Fig. 9, after  $\Delta t = 0.24T$  suddenly a large error accrues. It happened because of local instability between  $\Delta t = 0.24T$  and  $\Delta t = 0.5T$ . In this figure, for better comparison, a negative factor is multiplied in  $\tau$ 's.



**Fig. 7** Curves of relative period error versus the  $\Delta t/T$  for different values of  $\alpha$



**Fig. 8** Curves of algorithmic damping versus the  $\Delta t/T$  for different values of  $\alpha$



Figures 9 and 10 show relative period error (dispersion) and algorithmic damping (dissipation) of the method compared to Newmark, Wilson and Bathe methods. In this figure, ‘Alpha QB-Spline’ is the abbreviation standing for Alpha Quartic B-spline, i.e., proposed method. The parameters  $\alpha_1$  and  $\alpha_2$  of the proposed method have been considered in the minimum and maximum range of the previous investigation for stability and accuracy.

In general, the data in Figs. 9 and 10 show that for the integration methods, the magnitude of one or both error measurements usually increases with the time step  $\Delta t$ . Meanwhile, for the specified time step, the magnitude of both error measurements is greater for short-period SDOF systems than for long-period systems. As the graphs show, for this range of  $\Delta t/T$ , the proposed method, i.e., MQB-

spline, shows a low rate of errors for both types of errors. The magnitude of error for one or both kinds of errors for the proposed method is less than Bathe, Wilson and many cases of Newmark methods.

## 6 Numerical Evaluation

In this section, the validity of the proposed method is confirmed with the examination of several results. Four numerical examples are considered. The Bathe, Newmark and Wilson methods are employed for numerical simulation.

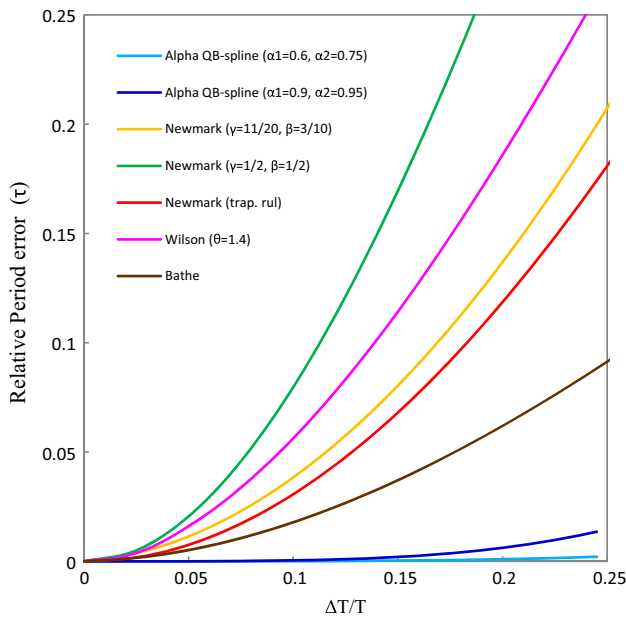


Fig. 9 Relative period error (dispersion)

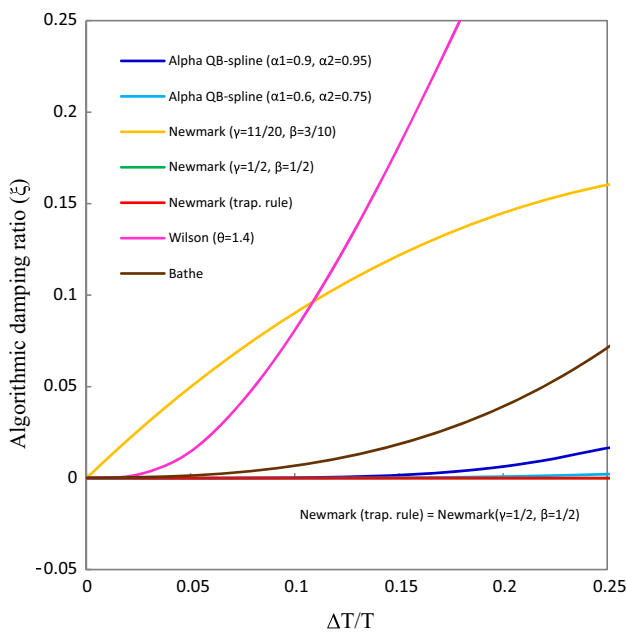


Fig. 10 Algorithmic damping (dissipation)

### 6.1 A SDOF System

A SDOF system used for numerical simulation is defined by the following equation:

$$u^{(2)}(t) + 1.2u^{(1)}(t) + 16\pi^2u(t) = \sin 3t, \quad u(0) = 0, \quad (62)$$

$$u^{(1)}(0) = 0$$

There is an exact solution for this equation (Paz 2012). To evaluate the accuracy of the proposed method, the

relative error of various methods is compared to exact solution. The time increments  $\Delta t$  used in this numerical study have been chosen as 0.1, 0.25 and 0.5 s in order to examine the effect of time step variations on the accuracy of the proposed method. Relative errors are defined by

$$e_i = \left| \frac{u_{\text{exact}}^{(i)} - u_{\text{num}}^{(i)}}{u_{\text{exact}}^{(i)}} \right| 100\% \quad (i = 0, 1, 2) \quad (63)$$

in which  $u_{\text{exact}}^{(i)}$  and  $u_{\text{num}}^{(i)}$  are the exact solution and the numerical result at any time instant, respectively. In solving this equation, the parameters  $\alpha_1$  and  $\alpha_2$  of the proposed method have been considered in two cases, i.e.,  $(\alpha_1 = 0.6, \alpha_2 = 0.75)$  and  $(\alpha_1 = 0.9, \alpha_2 = 0.95)$ .

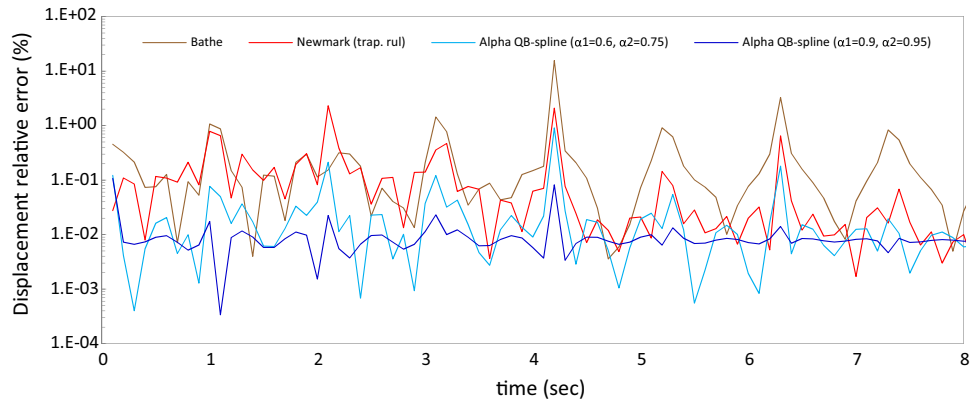
For this example, an accuracy analysis has been performed for 8, 20 and 40 s for time step equal to 0.1, 0.25 and 0.5 s, respectively. Figures 11, 12, 13, 14, 15, 16, 17, 18 and 19 show relative errors of displacement, velocity and acceleration for analysis with any time step. Of course, to present the figures much better, some results have been removed from them. The total average errors of this example (for all 80 time points) are also given in Table 3. The amount of this error is calculated by dividing the sum of errors which occur at each time instant by the number of all instants. It is clear from the table that for this example the proposed method gives a higher accuracy compared to the trapezoidal method.

### 6.2 Three-Degree-of-Freedom Spring System

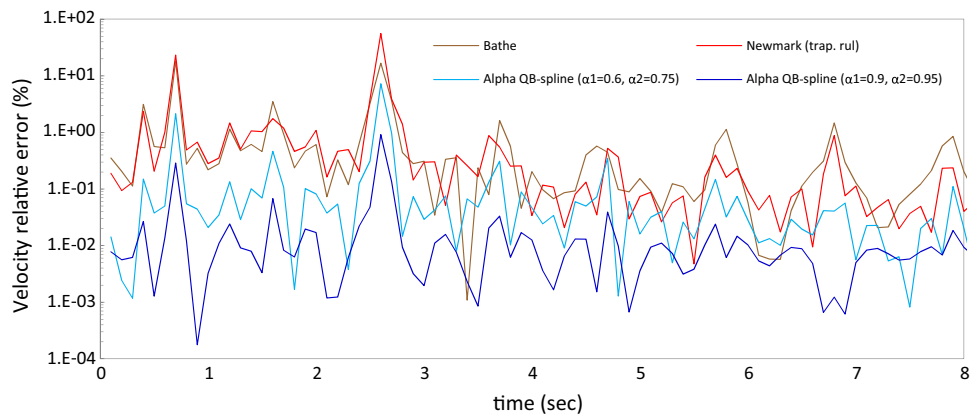
The objective of this section is to present the solution for a simple linear system that has been evaluated by Bathe and Noh (2012). The calculated solution shows the value of the method. The solution of the three-degree-of-freedom spring system is considered and shown in Fig. 20, for which node 1 is subjected to the prescribed displacement over time. The governing equation can be rewritten to solve only the unknown displacements  $u_2$  and  $u_3$ . For this system,  $k_1 = 10^7$ ,  $k_2 = 1$ ,  $m_1 = 0$ ,  $m_2 = 1$ ,  $m_3 = 1$  are considered and the displacement is prescribed at node 1 to be  $u_1 = \sin \omega_p t$  with  $\omega_p = 1.2$ .

The important point to note is that this simple problem as a ‘model problem’ represents the stiff and flexible parts of a much more complex structural system. In a mode superposition solution, the response within these stiff parts (a response that corresponds to very high artificial frequencies) would naturally not be included. In fact, the system shown in Fig. 20 is used as a ‘model system’ of such complex structural systems of many thousands of degrees of freedom and studying the behavior of the numerical solution when obtained by the direct integration method is desired. In this example, the spring system is considered using zero initial conditions for the

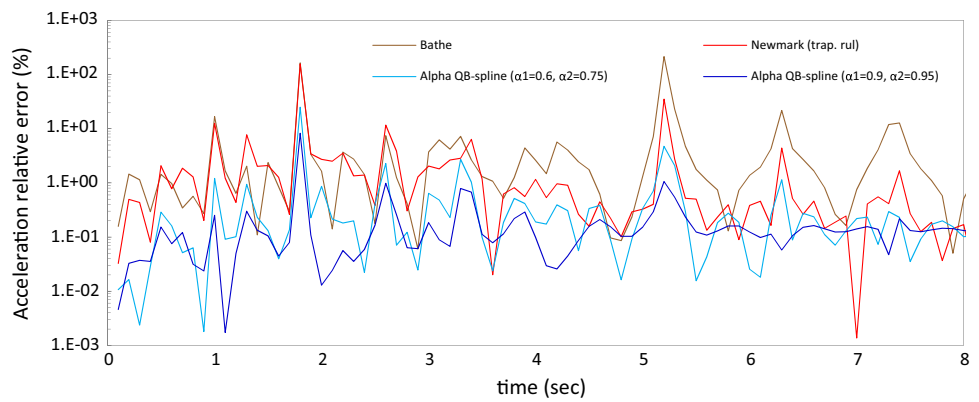
**Fig. 11** Displacement relative error for various methods ( $\Delta t = 0.1$ )



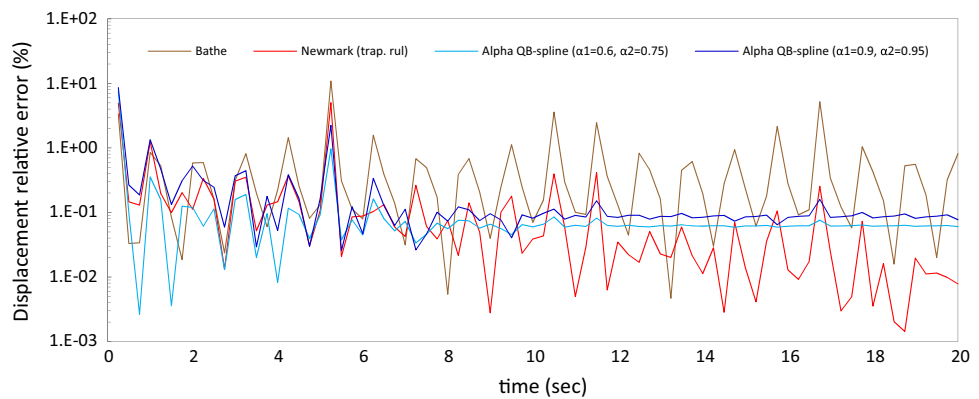
**Fig. 12** Velocity relative error for various methods ( $\Delta t = 0.1$ )



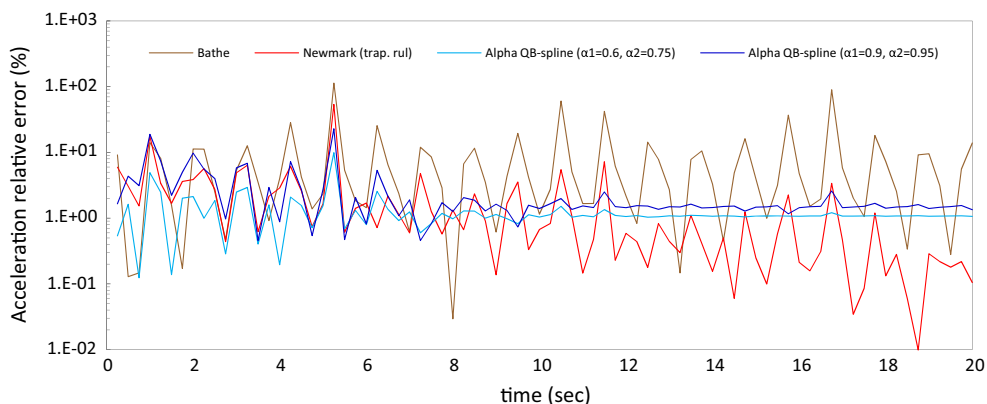
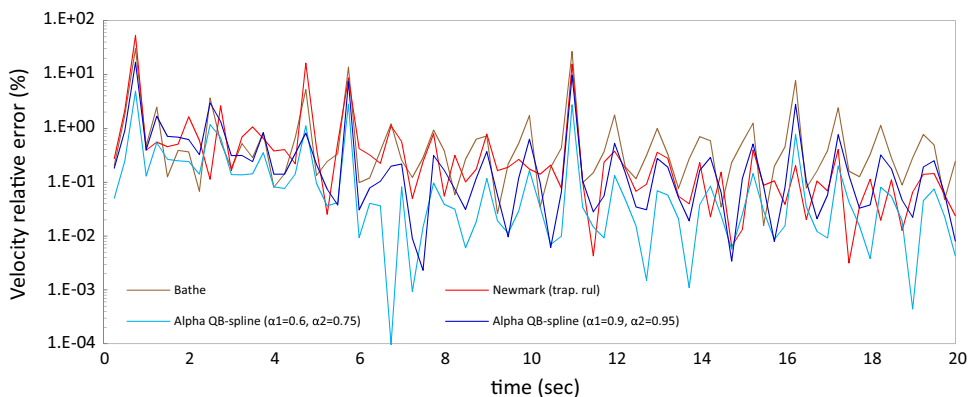
**Fig. 13** Acceleration relative error for various methods ( $\Delta t = 0.1$ )



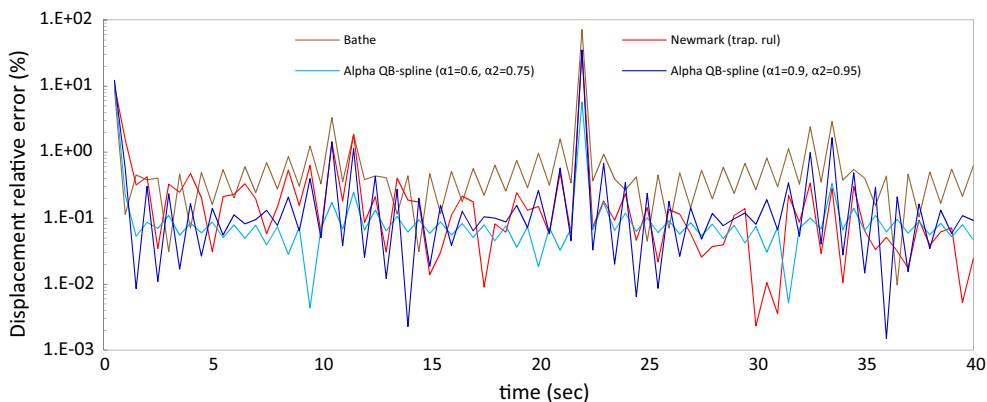
**Fig. 14** Displacement relative error for various methods ( $\Delta t = 0.25$ )



**Fig. 15** Velocity relative error for various methods ( $\Delta t = 0.25$ )



**Fig. 16** Acceleration relative error for various methods ( $\Delta t = 0.25$ )



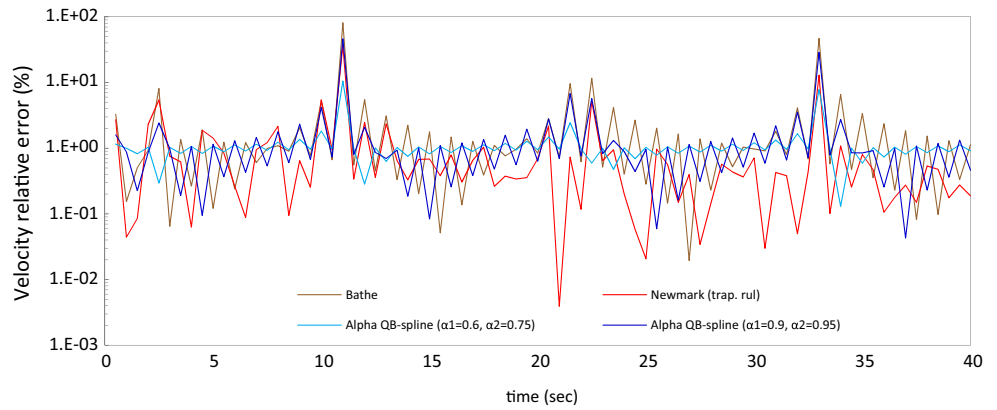
**Fig. 17** Displacement relative error for various methods ( $\Delta t = 0.5$ )

displacements and velocities at nodes 2 and 3 (as must typically be done in a complex many degrees of freedom structural analysis) and is solved for the response over 10 s. Time-stepping algorithm for the solution of this system is used. The time step used is  $\Delta t = 0.2618$ ; hence,  $\Delta t/T_1 = 0.0417$  and  $\Delta t/T_2 = 131.76$ , where  $T_1 = 6.283$ ,  $T_2 =$

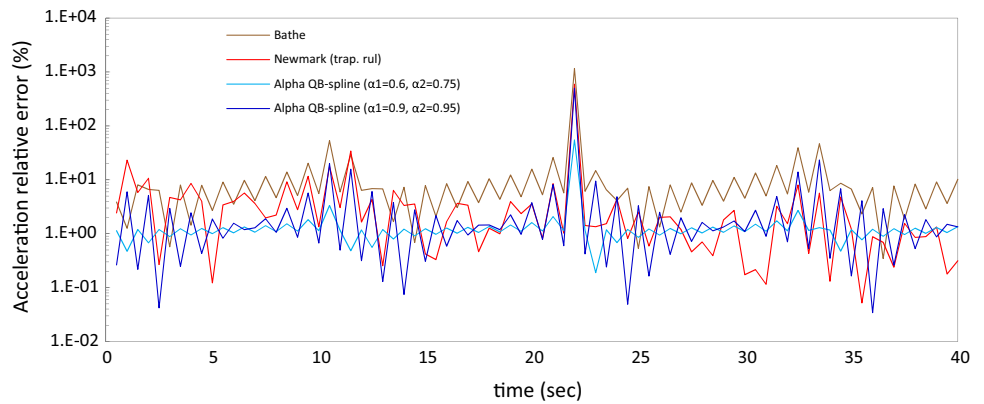
0.002 are the natural periods of the system with two degrees of freedom.

Figures 21, 22, 23, 24, 25 and 26 show the calculated solutions of node 2. Because for solutions of node 3, the responses of all numerical methods are too close; here, only node 2 is investigated. These figures also give the response obtained in a mode superposition solution,

**Fig. 18** Velocity relative error for various methods ( $\Delta t = 0.5$ )



**Fig. 19** Acceleration relative error for various methods ( $\Delta t = 0.5$ )



referred to as ‘reference solution’ using only the lowest frequency mode plus the static correction (Bathe and Noh 2012). For this example, only the case ( $\alpha_1 = 0.9, \alpha_2 = 0.95$ ) has been selected because for ( $\alpha_1 = 0.6, \alpha_2 = 0.75$ ) the solution is unstable.

The figures show that in all time-stepping schemes, the proposed and the Bathe methods perform very well; particularly, the velocity and acceleration at node 2 are very well predicted. But it should be noted that the Bathe method is a two-step method where for each time step, trapezoidal rule is used in the first half step and three-point

**Table 3** Error estimation analysis, for example 6.1

	Bathe	Wilson ( $\theta = 1.4$ )	Newmark (trap. rule)	Newmark ( $\gamma = 0.5$ $\beta = 0.5$ )	Newmark ( $\gamma = 0.55$ $\beta = 0.3$ )	Alpha QB-Spline ( $\alpha_1 = 0.6$ $\alpha_2 = 0.75$ )	Alpha QB-Spline ( $\alpha_1 = 0.9$ $\alpha_2 = 0.95$ )
$\Delta t = 0.1$							
Dis. err.	0.191	0.113	0.122	0.147	0.075	0.01	0.028
Vel. err.	0.529	0.847	1.142	1.654	0.936	0.023	0.154
Acc. err.	4.596	1.249	3.008	2.089	1.806	0.235	0.573
$\Delta t = 0.25$							
Dis. err.	0.389	0.314	0.222	0.236	0.116	0.274	0.176
Vel. err.	1.136	0.879	1.478	1.462	0.852	0.722	0.243
Acc. err.	5.518	0.518	2.376	2.454	0.820	2.595	1.321
$\Delta t = 0.5$							
Dis. err.	1.658	7.58	0.761	9.413	0.317	0.786	0.295
Vel. err.	1.422	3.774	1.395	56.217	0.615	2.051	1.191
Acc. err.	25.389	8.016	10.832	141.719	2.276	8.913	1.848

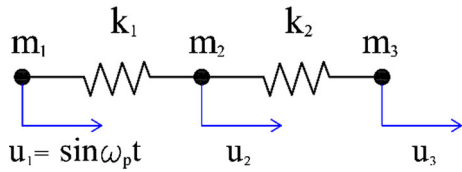


Fig. 20 Model problem of three-degree-of freedom spring system

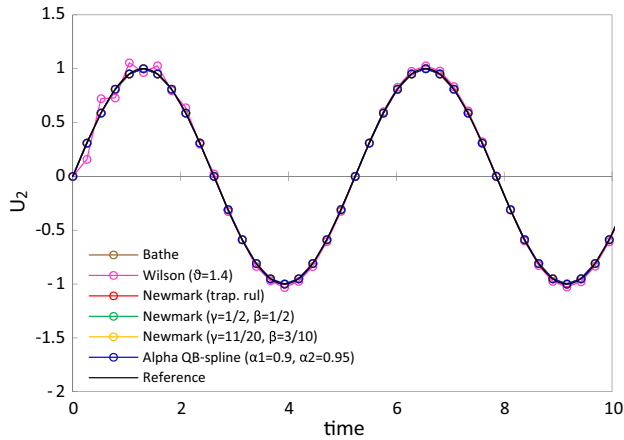


Fig. 21 Displacement of node 2 for different methods

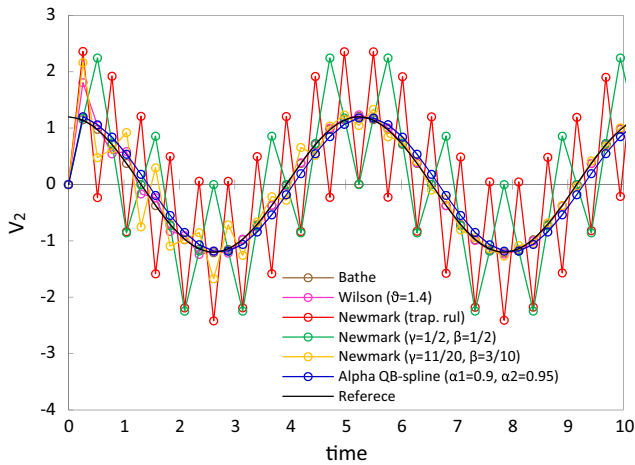


Fig. 22 Velocity of node 2 for different methods

backward difference method is used in the second half step (see Bathe 2007; Bathe and Noh 2012). However, all other methods under study in this paper are one step. In fact, the trapezoidal rule displays large errors and instability in the calculation of the acceleration at node 2; see Fig. 24. Some methods were omitted in Figs. 25 and 26 in order to make the comparison easier. In Fig. 26, the first step error in the proposed method is less than that in the Bathe method. Solving this example with the proposed method, unlike the basic method which was the subject of Rostami et al. (2012) and Shojaee et al. (2015), shows that the modified

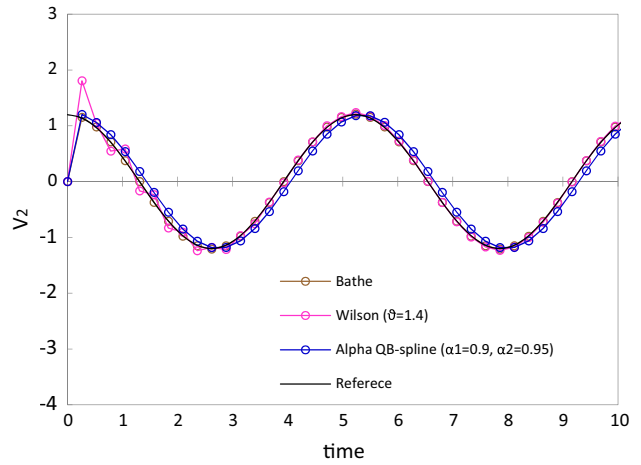


Fig. 23 Velocity of node 2 for different methods

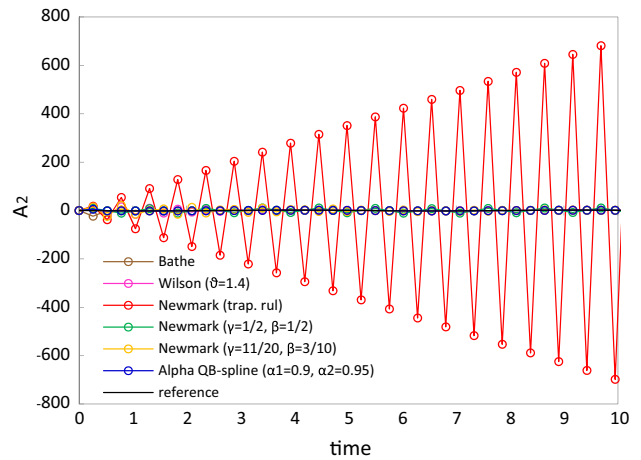


Fig. 24 Acceleration of node 2 for different methods

quartic B-spline method always maintains its stability and shows high accuracy.

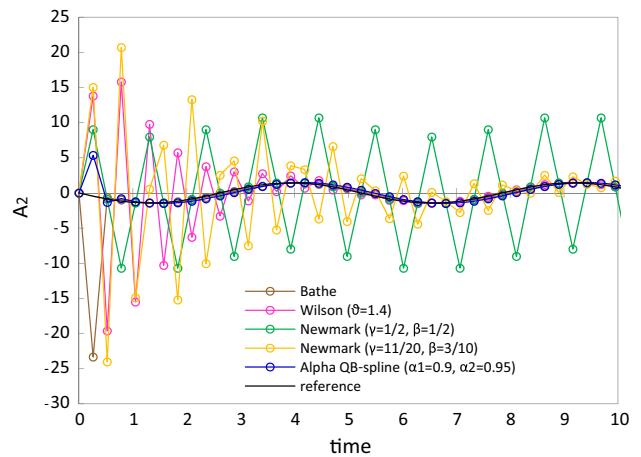


Fig. 25 Acceleration of node 2 for different methods



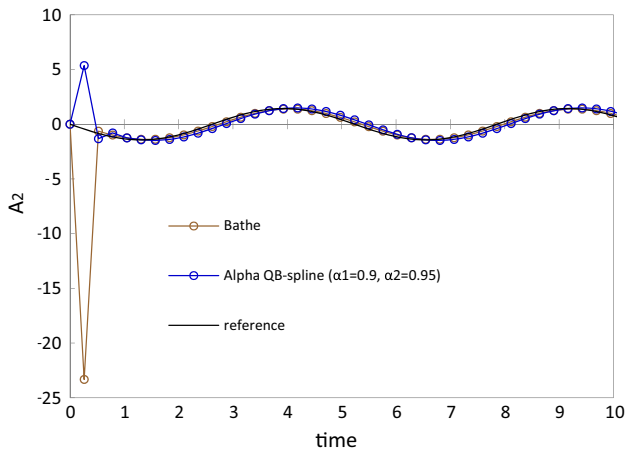
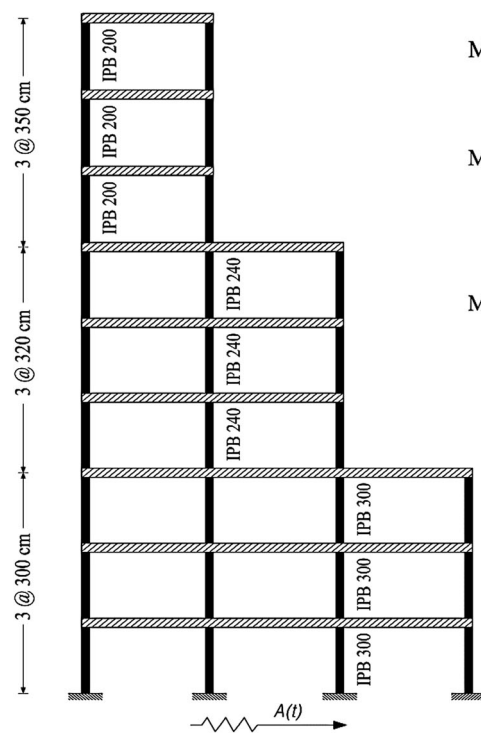


Fig. 26 Acceleration of node 2 for different methods

Fig. 27 Nine-story shear building subjected to harmonic ground acceleration

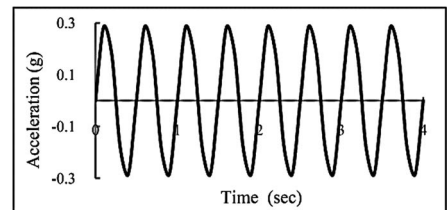


Modulus of Elasticity =  $2.1 \times 10^6 \text{ kg/cm}^2$

Moment of Inertia  $\begin{cases} \text{IPB300} = 25170 \text{ cm}^4 \\ \text{IPB240} = 11260 \text{ cm}^4 \\ \text{IPB200} = 5700 \text{ cm}^4 \end{cases}$

Mass  $\begin{cases} \text{Story 1 to 3} = 1300 \text{ kg} \\ \text{Story 4 to 6} = 850 \text{ kg} \\ \text{Story 7 to 9} = 500 \text{ kg} \end{cases}$

$A(t) = 0.3g \times \sin(4\pi t)$   
 $g = 981 \text{ cm/sec}^2$



Ground Acceleration

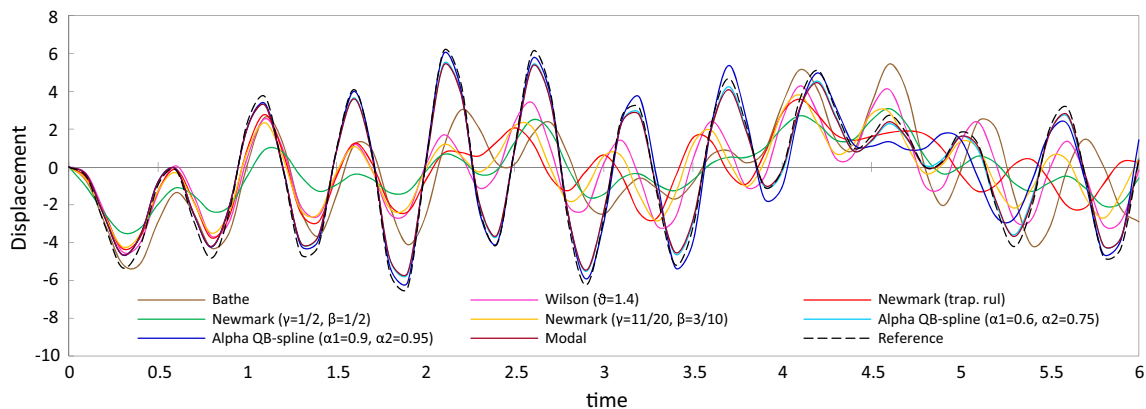


Fig. 28 Displacement–time history of story 1 in frame shown in Fig. 27

### 6.3 A Nine-Story Shear Building Under Harmonic Ground Acceleration

Figure 27 shows a nine-story shear building subjected to sinusoidal base acceleration with frequency of  $4\pi \text{ rad/s}$  and  $\text{PGA} = 0.3 \text{ g}$  (Rostami et al. 2012). The columns are I-shaped with sections of IPB240, IPB270 and IPB300 used in first to third three stories, respectively. The least period of this system is equal to 0.41 s. In this example, the damping effects are negligible and  $\Delta t = 0.1 \text{ s}$  has been selected as the time increment.

This example has been also analyzed by the other methods which were used in the previous example. Displacement–time, velocity–time and acceleration–time histories of stories two and nine have been plotted in time

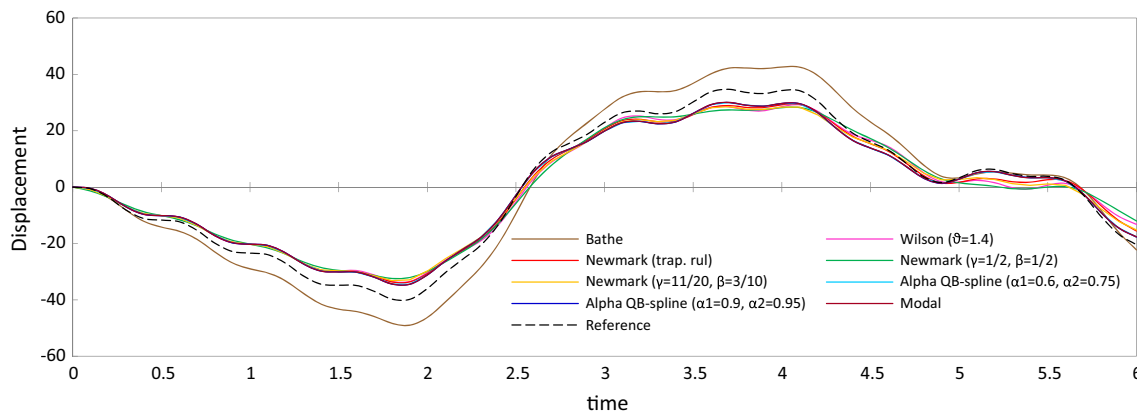


Fig. 29 Displacement–time history of story 9 in frame shown in Fig. 27

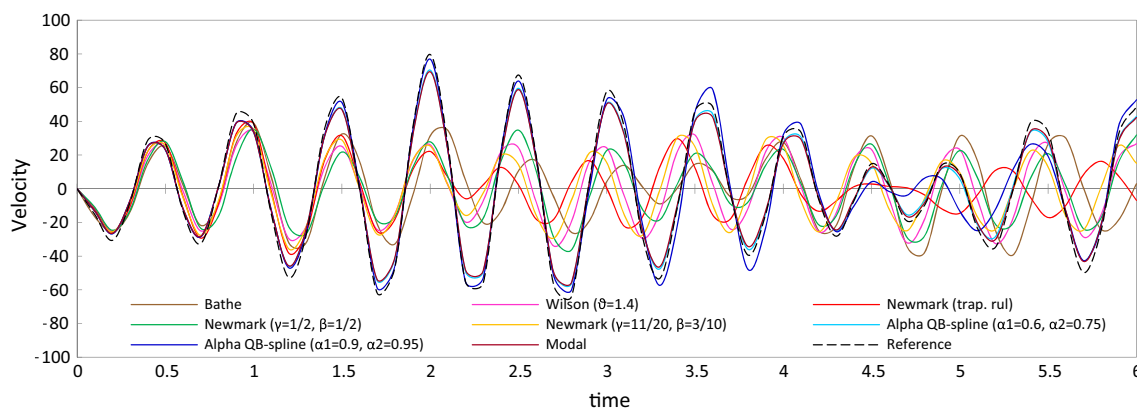


Fig. 30 Velocity–time history of story 1 in frame shown in Fig. 27

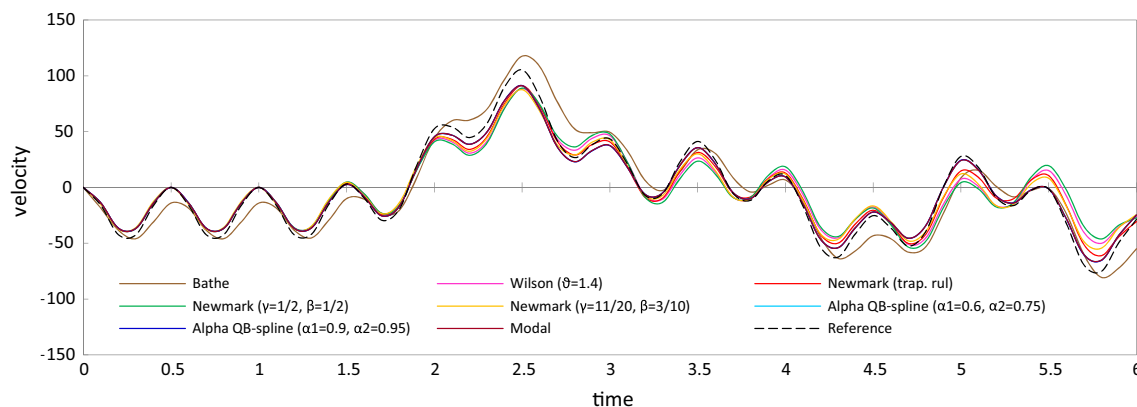


Fig. 31 Velocity–time history of story 9 in frame shown in Fig. 27

interval between 0 and 6 s in Figs. 28, 29, 30, 31, 32 and 33. In these graphs, the reference method is the same as mode superposition method in which the separated equations are being solved through the exact solution (response of SDOF systems to harmonic loads), while in the modal,

these equations are being solved through the numerical solution of Duhamel integration method. The graph shows that the result of the proposed method is closer to the exact solution than the other methods.

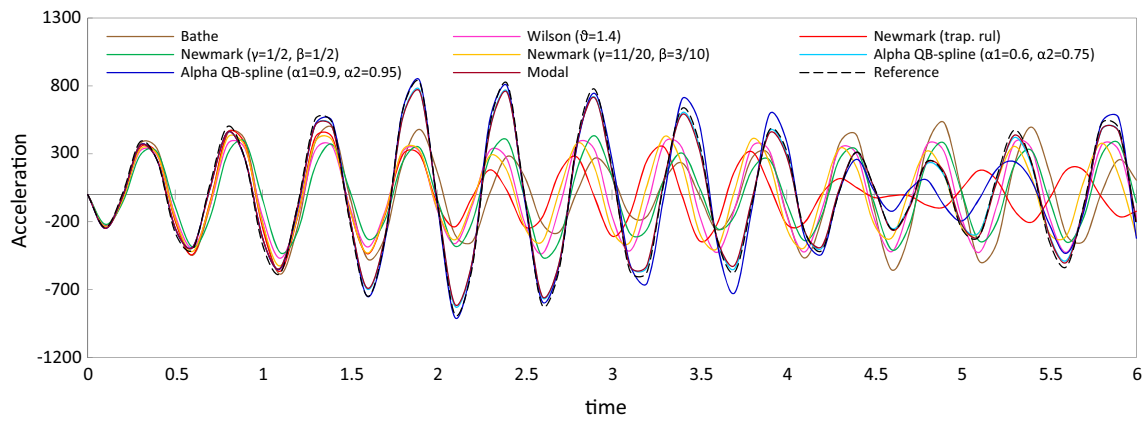


Fig. 32 Acceleration–time history of story 1 in frame shown in Fig. 27

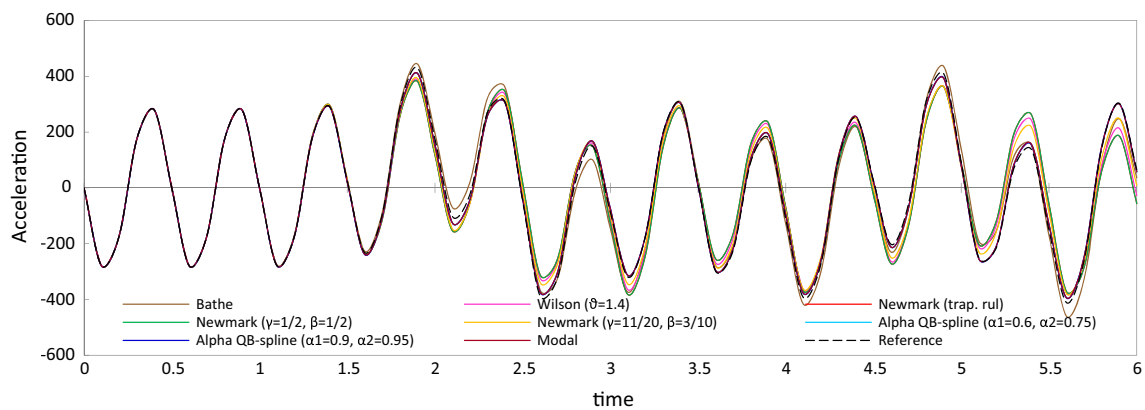


Fig. 33 Acceleration–time history of story 9 in frame shown in Fig. 27

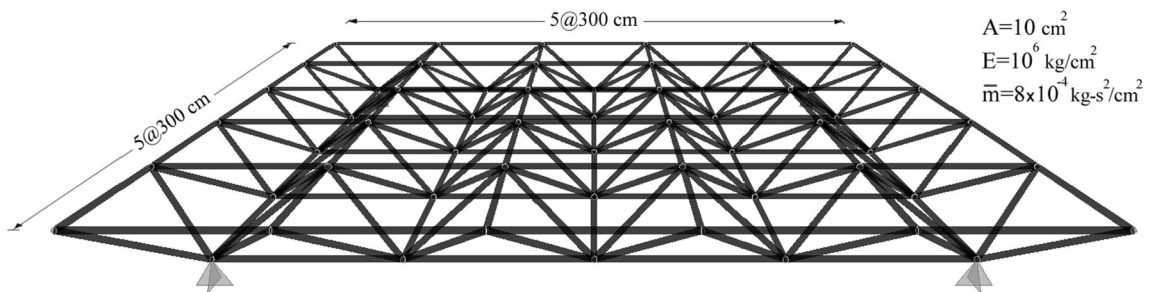


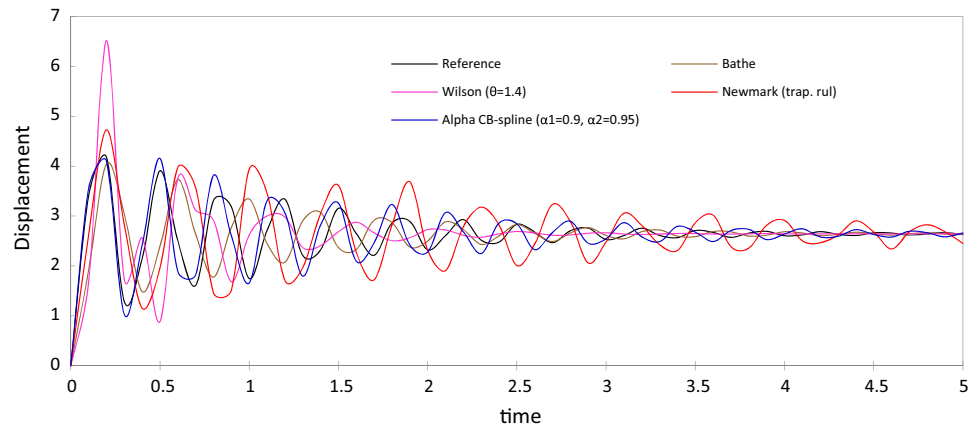
Fig. 34 Configuration and properties of double-layer spatial structure

### 6.4 Double-Layer Spatial Structure Under Step Loading

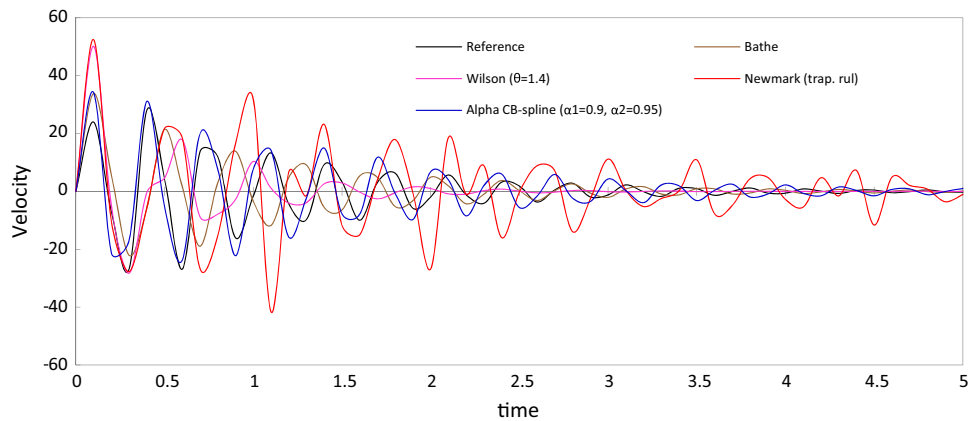
The configuration and dimensions of a structure shown in Fig. 34 are a 3-D view of a space truss retrieved from Rostami et al. (2012). The model is a double-layer spatial structure with four supports in the edges of the bottom layer. The total lengths of the lower layer and upper layer

are 1200 and 1500 cm in both directions, respectively. The height of the structure is 106 cm. This truss is composed of 200 members and 61 nodes. The module of elasticity, cross-sectional area and mass per length for all members are the same as what is shown in the figure. This structure is under four 3 ton concentrated impact loads, which suddenly strike on the nodes in the corners of central panel of the upper layer. In the analysis, the damping matrix is

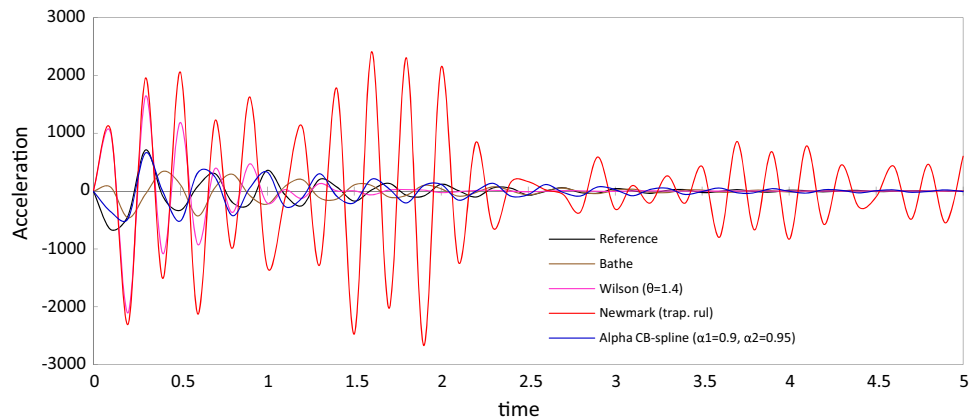
**Fig. 35** Vertical displacement–time history of the node in the middle of the top layer ( $\Delta t = 0.1$ )



**Fig. 36** Vertical velocity–time history of the node in the middle of the top layer ( $\Delta t = 0.1$ )



**Fig. 37** Vertical acceleration–time history of the node in the middle of the top layer ( $\Delta t = 0.1$ )

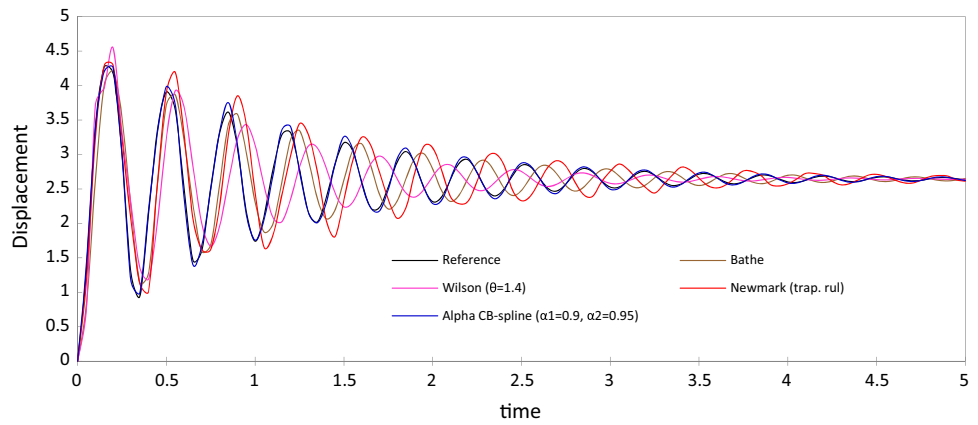


derived based on the mass matrix and the damping ratio of all 171 modes of the structure so that the equivalent viscous damping ratio is equal to 5% for all modes (Paz 2012).

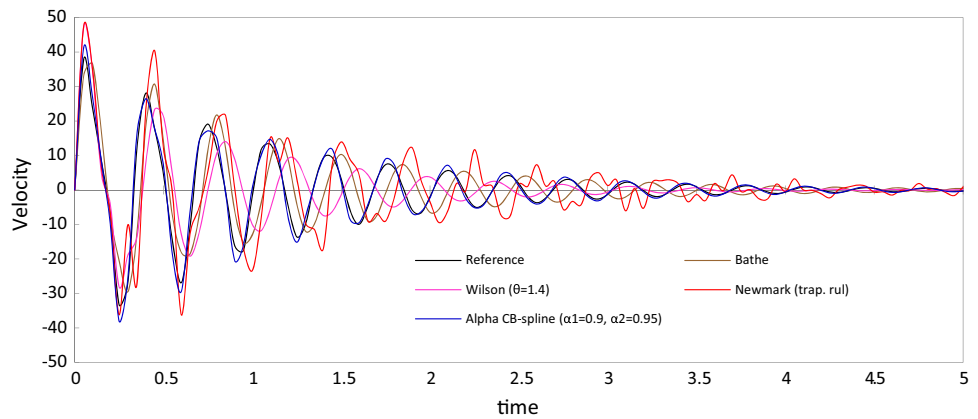
The maximum and minimum period of this system is 0.336 and 0.011 s, respectively. In order to investigate the accuracy of the proposed method with time consumption analysis, five different values 0.1, 0.05, 0.02, 0.01 and 0.005 s have been chosen as time step  $\Delta t$ .

This example is also analyzed by the methods which were used in previous examples. An accuracy analysis with computational time consumption has been performed by the use of a computer (Core i7 CPU @ 2.2 GHz). In this example, just trapezoidal rule from the Newmark method has been selected. The complete analysis has been done, and the results including the values of displacement, velocity and acceleration of all joints have been calculated.

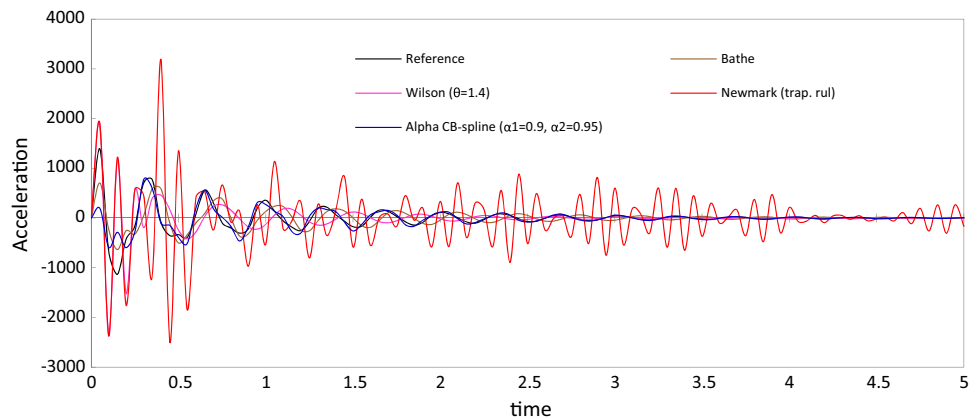
**Fig. 38** Vertical displacement–time history of the node in the middle of the top layer ( $\Delta t = 0.05$ )



**Fig. 39** Vertical velocity–time history of the node in the middle of the top layer ( $\Delta t = 0.05$ )



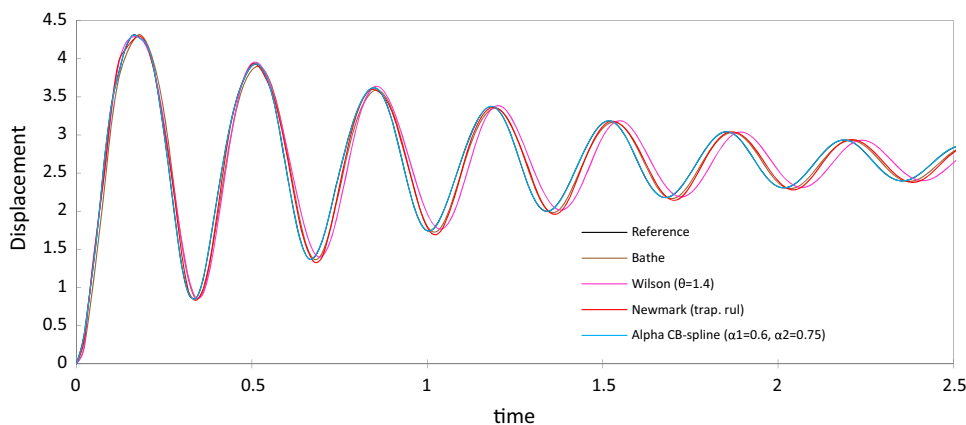
**Fig. 40** Vertical acceleration–time history of the node in the middle of the top layer ( $\Delta t = 0.05$ )



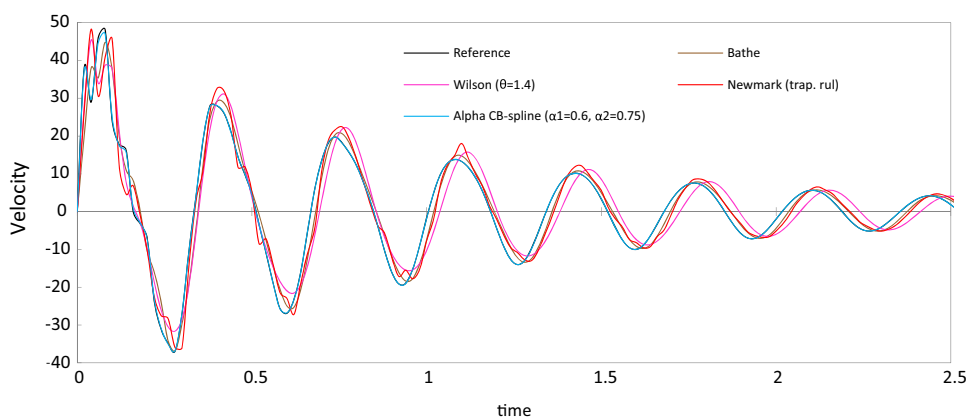
For example, the vertical displacement, velocity and acceleration of the node which is in the middle of the lower layer have been plotted as time history graphs. The ‘reference solution’ refers to mode superposition solution where all modes have been considered wherein and the separated equations are being solved through the Duhamel integral (Paz 2012). Figures 35, 36, 37, 38, 39, 40, 41, 42 and 43 show the vertical response of intended joint for time

step 0.02, 0.05 and 0.1 s. The figures show that in all time-stepping schemes, the proposed method performs well; particularly, the acceleration is very well predicted. Indeed, the trapezoidal rule displays large errors in the calculation of the acceleration; see Figs. 37, 40 and 43. In this example, for the time steps less than 0.02 s, collocation parameter is chosen as  $\alpha_1 = 0.6$  and  $\alpha_2 = 0.75$  because

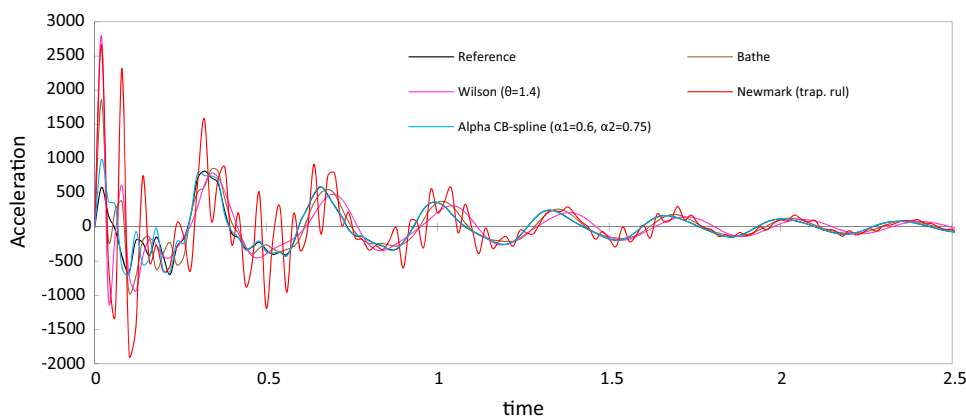
**Fig. 41** Vertical displacement–time history of the node in the middle of the top layer ( $\Delta t = 0.02$ )



**Fig. 42** Vertical velocity–time history of the node in the middle of the top layer ( $\Delta t = 0.02$ )



**Fig. 43** Vertical acceleration–time history of the node in the middle of the top layer ( $\Delta t = 0.02$ )



taking these parameters as  $\alpha_1 = 0.9$  and  $\alpha_2 = 0.95$  causes instability.

These analyses have been performed for 5 s after the impact loads being applied. The results of this analysis are demonstrated in Table 4. The normal root mean square error (NRMSE) method was used as error estimation criterion to investigate accuracy. The NRMSE represents the sample standard deviation of the differences between predicted values and observed values. These individual

differences are called residuals when the calculations are performed over the data sample used for estimation, and are called prediction errors when computed out-of-sample (Hyndman and Koehler 2006).

Here the NRMSE is defined as follows:

$$NRMSE = \frac{\sqrt{\sum_{t=1}^n (\hat{X}_t - X_t)^2 / n}}{X_t^{\max} - X_t^{\min}} \tag{64}$$



**Table 4** Error estimation and time consumption (s) analysis, for example 6.4

	Bathe	Wilson ( $\theta = 1.4$ )	Newmark (trap. rule)	Alpha QB-Spline
$\Delta t = 0.1$				
Dis. err.	0.629	0.72	0.817	0.278
Vel. err.	11.25	10.19	17.28	5.63
Acc. err.	242.6	480.3	1228.6	113.4
Time	0.0064	0.0037	0.0023	0.0069
$\Delta t = 0.05$				
Dis. err.	0.371	0.499	0.466	0.042
Vel. err.	6.83	8.91	9.8	1.49
Acc. err.	182.9	371	721	155.6
Time	0.0112	0.0065	0.0087	0.0156
$\Delta t = 0.02$				
Dis. err.	0.103	0.175	0.094	0.001
Vel. err.	2.75	4.02	3.34	0.16
Acc. err.	128.55	199.24	320.02	48.02
Time	0.029	0.012	0.028	0.037
$\Delta t = 0.01$				
Dis. err.	0.042	0.05	0.025	0.00
Vel. err.	1.25	1.49	1.26	0.06
Acc. err.	148.7	130.3	189.7	27.9
Time	0.083	0.036	0.054	0.106
$\Delta t = 0.005$				
Dis. err.	0.019	0.014	0.007	0.00
Vel. err.	0.61	0.67	0.55	0.00
Acc. err.	91.85	95.28	109.84	1.97
Time	0.15	0.075	0.085	0.19

where  $\hat{X}_t$  and  $X_t$  are the reference and resulted solution values (displacement, velocity or acceleration) at time  $t$ , respectively.  $n$  is the number of time steps.

Table 4 shows the response computational error analysis. This table shows NRMS error for various time steps. It is clear from the table that the proposed method gives a higher accuracy compared to the others. In this table, the results of time consumption analysis are shown as well.

## 7 Concluding Remarks

A new version of quartic B-spline direct time integration method for dynamic analysis of structures is presented. This procedure is derived based on the uniform quartic B-spline piecewise polynomial approximations and collocation method, named *Alpha Quartic B-Spline method*. By using two collocation parameters  $\alpha_1$  and  $\alpha_2$  in the proposed algorithm, unconditional stability was achieved, but a local instability was created. A stability analysis showed that unconditional stability is provided when a suitable value of  $\alpha_s$  is selected. The accuracy analysis indicated that the scheme benefits from high accuracy with low amplitude

decay and period elongation. Thus, it leads to a lower numerical amplitude dissipation and period dispersion compared to Wilson and Newmark trapezoidal rule. The effectiveness and robustness of the proposed algorithm in solving linear dynamic problems were demonstrated in the numerical examples. These evaluations showed that the proposed method operates more effectively than the trapezoidal rule and works as good as the Bathe method. The new implicit scheme is simple, effective and practical when the trapezoidal rule is not effective or even fails.

## References

- Bathe K-J (1996) Finite element procedures. Prentice-Hall, Englewood Cliffs, New Jersey
- Bathe K-J (2007) Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Comput Struct* 85(7):437–445
- Bathe K-J, Noh G (2012) Insight into an implicit time integration scheme for structural dynamics. *Comput Struct* 98:1–6
- De Boor C, De Boor C, Mathématicien E-U, De Boor C, De Boor C (1978) A practical guide to splines. Springer, New York

- Dokainish M, Subbaraj K (1989) A survey of direct time-integration methods in computational structural dynamics—I. Explicit methods. *Comput Struct* 32(6):1371–1386
- Hilber HM, Hughes TJ (1978) Collocation, dissipation and [over-shoot] for time integration schemes in structural dynamics. *Earthq Eng Struct Dyn* 6(1):99–117
- Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. *Int J Forecast* 22(4):679–688
- Lu T-T, Shiou S-H (2002) Inverses of  $2 \times 2$  block matrices. *Comput Math Appl* 43(1–2):119–129
- Malakiyeh MM, Shojaee S, Javaran SH (2018) Development of a direct time integration method based on Bezier curve and 5th-order Bernstein basis function. *Comput Struct* 194:15–31
- Newmark NM (1959) A method of computation for structural dynamics. *J Eng Mech Div* 85(3):67–94
- Noh G, Bathe K-J (2013) An explicit time integration scheme for the analysis of wave propagations. *Comput Struct* 129:178–193
- Paz M (2012) *Structural dynamics: theory and computation*. Springer, Berlin
- Piegl L, Tiller W (2012) *The NURBS book*. Springer, Berlin
- Rogers D (2001) *An introduction to NURBS: with historical perspective*. Morgan Kaufmann, San Francisco
- Rostami S, Shojaee S (2017) Alpha-modification of cubic B-spline direct time integration method. *Int J Struct Stab Dyn*. <https://doi.org/10.1142/S0219455417501188>
- Rostami S, Shojaee S (2018) A family of cubic B-spline direct integration algorithms with controllable numerical dissipation and dispersion for structural dynamics. *Iran J Sci Technol Trans Civ Eng* 42(1):17–32
- Rostami S, Shojaee S, Moeinadini A (2012) A parabolic acceleration time integration method for structural dynamics using quartic B-spline functions. *Appl Math Model* 36(11):5162–5182
- Rostami S, Shojaee S, Saffari H (2013) An explicit time integration method for structural dynamics using cubic B-spline polynomial functions. *Sci Iran* 20(1):23–33
- Shojaee S, Rostami S, Moeinadini A (2011) The numerical solution of dynamic response of SDOF systems using cubic B-spline polynomial functions. *Struct Eng Mech* 38(2):211–229
- Shojaee S, Rostami S, Abbasi A (2015) An unconditionally stable implicit time integration algorithm: modified quartic B-spline method. *Comput Struct* 153:98–111
- Subbaraj K, Dokainish M (1989) A survey of direct time-integration methods in computational structural dynamics—II. Implicit methods. *Comput Struct* 32(6):1387–1401
- Wen WB, Luo SM, Jian KL (2015) A novel time integration method for structural dynamics utilizing uniform quintic B-spline functions. *Arch Appl Mech* 85(12):1743–1759
- Wen W, Duan S, Yan J, Ma Y, Wei K, Fang D (2017) A quartic B-spline based explicit time integration scheme for structural dynamics with controllable numerical dissipation. *Comput Mech* 59(3):403–418
- Wilson EL (1968) *A computer program for the dynamic stress analysis of underground structures*. California Univ Berkeley Structural Engineering Lab, Berkeley
- Wilson E, Farhoomand I, Bathe K (1972) Nonlinear dynamic analysis of complex structures. *Earthq Eng Struct Dyn* 1(3):241–252