OPEN ACCESS
International Journal of Computational Methods
Vol. 19, No. 10 (2022) 2250022 (33 pages)

DOI: 10.1142/S0219876222500220

World Scientific
www.worldscientific.com

Damping Perturbation Based Time Integration Asymptotic Method for Structural Dynamics

Mario Lázaro

Department of Continuum Mechanics and Theory of Structures
Universitat Politècnica de València, Spain

Instituto Universitario de Matemática Pura y Aplicada
Universitat Politècnica de València, Spain
malana@mes.upv.es

Received 21 February 2022
Accepted 10 May 2022
Published 24 June 2022

In this paper a novel time integration numerical method based on artificial perturbation of damping is proposed. Viscous dissipative terms in the structural dynamics equations of motion are perturbed by an artificial parameter. The subsequent asymptotic expansion of the transient response results in an infinite series which can be summed, leading to a well-defined new step-by-step explicit iterative scheme. Precise integration algorithms are designed for the construction of the main matrices. Conditions for convergence and numerical properties, i.e., stability and accuracy are also studied in detail. The proposed approach is validated with a numerical example, showing high accuracy with respect to other existing methods in literature.

Keywords: Time integration; damping perturbation; asymptotic method; structural dynamics; explicit algorithm; transient problem.

1. Introduction

In this paper, dynamic analysis of linear structures with time invariant properties and viscous damping is considered. Equilibrium of inertia, elastic and damping forces leads to well-known system of second-order differential equations

$$\mathbf{M}\,\ddot{\mathbf{u}} + \mathbf{C}\,\dot{\mathbf{u}} + \mathbf{K}\,\mathbf{u} = \mathbf{f}(t), \quad \mathbf{u}(0) = \mathbf{u}_0, \quad \dot{\mathbf{u}}(0) = \dot{\mathbf{u}}_0, \tag{1}$$

where $\mathbf{M}, \mathbf{C}, \mathbf{K} \in \mathbb{R}^{N\times N}$ are the mass, damping and stiffness matrices, respectively, and $\mathbf{f}(t) \in \mathbb{R}^N$ represents the time-dependent vector of external forces at time t. The mass matrix is assumed to be symmetric and positive definite, while $\mathbf{K}$ and $\mathbf{C}$ are positive semidefinite symmetric matrices. The N degrees of freedom (dof) are arranged in the column vector $\mathbf{u}(t) \in \mathbb{R}^N$.

Explicit methods allow calculation of the solution at each time step from the previous time step using simple matrix product. However, a known drawback of explicit methods is their conditional stability, such as the central difference method (CDM) or the fourth-order Runga–Kutta method (RK4) [Burden and Faires (2001); Butcher (2008)]. This is partially overcome by the family of precise integration methods (PIM) of special interest in linear models of structural dynamics. The main contribution of the PIM, originally proposed by Zhong and Williams [1994] Wan-Xie [2004], is the proposal of an explicit algorithm based on the precise computation of the exponential matrix by means of the 2^p algorithm in combination with the evaluation of the incremental part [Fung (2004)]. The method exhibits conditional stability although within such a wide range of time steps that in practice it can be considered unconditionally stable. In addition, it shows highly precise results in the homogeneous problem, although the accuracy drops when there exist nonhomogeneous terms in the equations. Zhong and Williams [1994] proposed to linearly interpolate the external applied force in the time step, which needs the computation of an inverse matrix with the consequent loss of accuracy. Since the first attempt of Zhong and Williams, a variety of proposals to improve the efficiency and accuracy of the method have been published. Lin *et al.* [1995] improved the PIM by developing in trigonometric Fourier series the external force term within the time step. Wang and Zhou [2005] proposed the so-called Modified Precise Integration Method (MPIM), which consisted of evaluating the nonhomogeneous term by using Gaussian quadrature integration. The approach allows achieving high accuracy in the result in exchange for increasing the computational cost. Wang and Au [2007] presented a PIM avoiding computation of the inverse matrix for the evaluation of the nonhomogeneous terms. They proposed two methods for the evaluation of the independent term based on interpolation with Chebyshev [Wang and Au (2008)] and Lagrange polynomials [Wang and Au (2009)]. Fung [1997] used the principle of precise integration, combining the 2^p algorithm with the evaluation of the incremental part, to determine the step-response and impulse-response matrices of the system, as well as their derivatives. Fung and Chen [2006] also proposed an efficient algorithm based on the precise integration, using the Krylov subspace and the Padé approximations. In the work of Zhu and Law [2001], the PIM is applied for a continuous Euler–Bernouilli beam under moving loads. Caprani [2013] developed a PIM based on modal analysis for moving forces in footbridge vibration response.

Perturbation techniques have been applied for the numerical solution of linear and nonlinear, algebraic and differential equations, by expanding the solution in terms of certain parameter which is known *a prior* to be very small with respect to other terms of the model [Nayfeh (2004)]. Real structures are usually assumed to be lightly damped, that is, dissipative forces can be considered much smaller than inertia and elastic forces. Artificial perturbation of dissipative terms in linear structural dynamics has been successfully applied mainly for the computation of complex frequencies and modes of nonclassically viscously [Meirovitch and Ryland (1979, 1985); Chung and Lee (1986); Peres-Da-Silva *et al.* (1995); Cha (2005); Lázaro

(2016)] and nonviscously [Daya and Potier-Ferry (2001); Duigou *et al.* (2003); Lázaro and Pérez-Aparicio (2013); Zoghaib and Mattei (2014); Lázaro (2015,2018); Lázaro *et al.* (2016)] damped structures. In the time domain, techniques based on asymptotic perturbation are very useful to obtain solutions in nonlinear mechanics [Gallagher (1975); Cochelin *et al.* (1994); Mei *et al.* (2008)]. In linear structural dynamics, Fafard *et al.* [1997] and Berrahma-Chekroun *et al.* [2001] have developed asymptotic methods by expanding the transient response in time power series. However, hardly any references can be found in the literature on asymptotic time-integration methods for the linear model of Eq. (1). This is maybe because they intrinsically involve two iterative processes: one advancing in the time domain and the other one along the asymptotic dimension. This fact makes them uncompetitive with other explicit methods that only involve step-by-step iterations in time.

In this paper, an asymptotic time integration algorithm based on artificial perturbation of damping for linear structural dynamics is proposed. A step-by-step explicit algorithm arises after the sum of the infinite series resulting from asymptotic expansion of the response, avoiding consequently a double iteration. Stability, conditions for convergence of the method will be explored and algorithms for the computation of the main matrices of the scheme will be developed. The accuracy of the proposed method is validated through a numerical example. Dependency of error with damping level and time step size are evaluated and compared to other implicit and explicit methods. In the usual ranges of damping present in real structures, the proposed approach shows highly accurate results with respect to the other existing methods.

2. Homotopy Analysis Based on Artificial Perturbation of Damping

Most problems of structural dynamics are governed by weak dissipation mechanisms so that the viscous damping forces are generally small compared to the inertial and elastic forces. Mathematically, viscously damped terms $\mathbf{C}\dot{\mathbf{u}}$ of Eq. (1) can be considered as a perturbation of the undamped case, something that might be reproduced with an artificial parameter. We propose to modify Eq. (1) with the paramete, say ϵ with $0 \leq \epsilon \leq 1$, which multiplies the damping matrix in Eq. (1) resulting

$$\mathbf{M}\ddot{\mathbf{u}} + \epsilon\,\mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}(t). \tag{2}$$

The solution of Eq. (2) is then a two-variable dependent array $\mathbf{u}(t,\epsilon)$. The ϵ-dependent solution can then be expanded as

$$\mathbf{u}(t,\epsilon) = \sum_{n=0}^{\infty} \mathbf{x}^{(n)}(t)\,\epsilon^n = \mathbf{x}^{(0)}(t) + \mathbf{x}^{(1)}(t)\,\epsilon + \mathbf{x}^{(2)}(t)\,\epsilon^2 + \cdots, \tag{3}$$

where $\{\mathbf{x}^{(n)}(t)\}_{n=0}^{\infty} \in \mathbb{R}^N$ is a sequence of functions to be determined. If series (3) is convergent for $0 \leq \epsilon \leq 1$ then the solution of our problem can be written as

$$\mathbf{u}(t) = \mathbf{u}(t,1) = \sum_{n=0}^{\infty} \mathbf{x}^{(n)}(t) = \mathbf{x}^{(0)}(t) + \mathbf{x}^{(1)}(t) + \mathbf{x}^{(2)}(t) + \cdots. \tag{4}$$

while the lower limit, $\epsilon = 0$, corresponds to the undamped problem,

$$\mathbf{u}(t, 0) = \mathbf{x}^{(0)}(t). \tag{5}$$

The initial conditions are satisfied by the first iteration $\mathbf{x}^{(0)}(t)$, so that $\mathbf{x}^{(0)}(0) = \mathbf{u}_0$ and $\dot{\mathbf{x}}^{(0)}(0) = \dot{\mathbf{u}}_0$. Thus, the problem for $n = 0$ can be written as

$$\begin{cases} \mathbf{M}\ddot{\mathbf{x}}^{(0)} + \mathbf{K}\,\mathbf{x}^{(0)} = \mathbf{f}(t), \\ \mathbf{x}^{(0)}(0) = \mathbf{u}_0, \quad \dot{\mathbf{x}}^{(0)}(0) = \dot{\mathbf{u}}_0. \end{cases} \tag{6}$$

Substituting Eq. (3) into Eq. (2) and identifying the coefficients of ϵ^n, the nth function $\mathbf{x}^{(n)}(t)$ can be determined. After some straight operations it yields

$$\begin{cases} \mathbf{M}\ddot{\mathbf{x}}^{(n)} + \mathbf{K}\,\mathbf{x}^{(n)} = -\mathbf{C}\,\dot{\mathbf{x}}^{(n-1)}, \\ \mathbf{x}^{(n)}(0) = \mathbf{0}, \quad \dot{\mathbf{x}}^{(n)}(0) = \mathbf{0}, \quad n \geq 1, \end{cases} \tag{7}$$

where homogeneous initial conditions have been imposed for $n \geq 1$ because the real ones have already been applied for $n = 0$ in Eq. (6). Let us see in detail the derivation of the resulting double iterative scheme, i.e., in the time domain and in the domain of the asymptotic expansion terms.

2.1. *Solution of the first iteration, $n = 0$*

The time domain will be sampled by the time-step Δt, i.e., $\{t_k,\ k \geq 0\}$, with $t_0 = 0$ and $t_{k+1} = t_k + \Delta t$. The first iteration ($n = 0$) at $t = t_k$ will be denoted by $\mathbf{x}_k^{(0)} = \mathbf{x}^{(0)}(t_k)$. A recursive scheme is built using the explicit method based on Green's functions [Mansur *et al.* (2007)] yielding

$$\begin{aligned} \mathbf{x}_{k+1}^{(0)} &= \mathbf{G}(\Delta t)\,\mathbf{x}_k^{(0)} + \mathbf{H}(\Delta t)\,\dot{\mathbf{x}}_k^{(0)} + \int_{t=t_k}^{t_{k+1}} \mathbf{H}(t_{k+1} - t)\mathbf{M}^{-1}\mathbf{f}(t)\,dt, \\ \dot{\mathbf{x}}_{k+1}^{(0)} &= \dot{\mathbf{G}}(\Delta t)\,\mathbf{x}_k^{(0)} + \dot{\mathbf{H}}(\Delta t)\,\dot{\mathbf{x}}_k^{(0)} + \int_{t=t_k}^{t_{k+1}} \mathbf{G}(t_{k+1} - t)\mathbf{M}^{-1}\mathbf{f}(t)\,dt. \end{aligned} \tag{8}$$

The Green's functions, $\mathbf{G}(t)$ and $\mathbf{H}(t)$, are the solutions of the following system of matrix differential equations:

$$\begin{cases} \dot{\mathbf{G}}(t) = -\mathbf{A}\,\mathbf{H}(t), \quad \mathbf{H}(0) = \mathbf{0}, \\ \dot{\mathbf{H}}(t) = \mathbf{G}(t), \quad \mathbf{G}(0) = \mathbf{I}_N, \end{cases}$$

where $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$ and $\mathbf{I}_N$ denotes the identity matrix of order N. It is straightforward that both $\mathbf{G}(t)$ and $\mathbf{H}(t)$ are solutions of the second-order matrix differential equations $\mathbf{M}\ddot{\mathbf{Z}} + \mathbf{K}\mathbf{Z} = \mathbf{0}$, where $\mathbf{Z}(t) \in \mathbb{R}^{N\times N}$ and hence both functions can be written as

$$\begin{aligned} \mathbf{G}(t) &= \cos(\sqrt{\mathbf{A}}\,t) = \sum_{j=0}^{\infty} \frac{(-1)^j\,\mathbf{A}^j}{(2j)!} t^{2j}, \\ \mathbf{H}(t) &= \mathbf{A}^{-1/2}\,\sin(\sqrt{\mathbf{A}}\,t) = \sum_{j=0}^{\infty} \frac{(-1)^j\,\mathbf{A}^j}{(2j+1)!} t^{2j+1}. \end{aligned} \tag{9}$$

Sometimes $\mathbf{G}(t)$ and $\mathbf{H}(t)$ are named step-response and impulsive-response functions [Fung (1997)], respectively. In order to obtain a closed form of the recursive scheme, the integrals of the nonhomogeneous part must be evaluated in the interval $[t_k, t_{k+1}]$. For that, we will assume a cubic interpolation of the external applied force using third-order Lagrange polynomials, namely

$$\mathbf{f}(t) \approx \sum_{i=1}^{4} \mathcal{L}_i \left(\frac{t - t_k}{\Delta t} \right) \mathbf{f} \left(t_k + \frac{(i-1)\Delta t}{3} \right), \quad t_k \le t \le t_{k+1}, \tag{10}$$

where the interpolation polynomials in terms of the parameter $\xi = (t - t_k)/\Delta t \in [0, 1]$ are

$$\begin{aligned} \mathcal{L}_1(\xi) &= \frac{1}{2}(1-\xi)(2-3\xi)(1-3\xi), \quad \mathcal{L}_2(\xi) = \frac{9}{2}(1-\xi)(2-3\xi)\xi, \\ \mathcal{L}_3(\xi) &= -\frac{9}{2}(1-\xi)(1-3\xi)\xi, \quad \mathcal{L}_4(\xi) = \frac{1}{2}(1-3\xi)(2-3\xi)\xi. \end{aligned} \tag{11}$$

Plugging Eq. (10) into the integrals of Eq. (8) and after some operations we obtain

$$\begin{aligned} \int_{t=t_k}^{t_{k+1}} \mathbf{H}(t_{k+1} - t)\mathbf{M}^{-1}\mathbf{f}(t)\, dt &\approx \sum_{i=1}^{4} \mathbf{L}_{ui}\, \mathbf{M}^{-1}\, \mathbf{f} \left(t_k + \frac{(i-1)\Delta t}{3} \right), \\ \int_{t=t_k}^{t_{k+1}} \mathbf{G}(t_{k+1} - t)\mathbf{M}^{-1}\mathbf{f}(t)\, dt &\approx \sum_{i=1}^{4} \mathbf{L}_{vi}\, \mathbf{M}^{-1}\, \mathbf{f} \left(t_k + \frac{(i-1)\Delta t}{3} \right), \end{aligned} \tag{12}$$

where

$$\begin{aligned} \mathbf{L}_{ui} &= \int_{t=t_k}^{t_{k+1}} \mathbf{H}(t_{k+1} - t)\, \mathcal{L}_i \left(\frac{t - t_k}{\Delta t} \right) dt, \\ \mathbf{L}_{vi} &= \int_{t=t_k}^{t_{k+1}} \mathbf{G}(t_{k+1} - t)\, \mathcal{L}_i \left(\frac{t - t_k}{\Delta t} \right) dt, \quad 1 \le i \le 4. \end{aligned} \tag{13}$$

Rearranging together displacements and velocities of the dof of the same $2N$-size vector, the iterative process can be summarized as the following relationship:

$$\mathbf{X}_{k+1}^{(0)} = \mathbf{T}\, \mathbf{X}_k^{(0)} + \mathbf{L}\, \mathbf{g}_k, \quad k \ge 0, \tag{14}$$

where

$$\mathbf{X}_k^{(0)} = \begin{Bmatrix} \mathbf{x}_k^{(0)} \\ \dot{\mathbf{x}}_k^{(0)} \end{Bmatrix} \in \mathbb{R}^{2N}, \quad \mathbf{X}_0^{(0)} = \begin{Bmatrix} \mathbf{u}_0 \\ \dot{\mathbf{u}}_0 \end{Bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{G}(\Delta t) & \mathbf{H}(\Delta t) \\ \dot{\mathbf{G}}(\Delta t) & \dot{\mathbf{H}}(\Delta t) \end{bmatrix} \in \mathbb{R}^{2N \times 2N}, \tag{15}$$

and

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{u1} & \mathbf{L}_{u2} & \mathbf{L}_{u3} & \mathbf{L}_{u4} \\ \mathbf{L}_{v1} & \mathbf{L}_{v2} & \mathbf{L}_{v3} & \mathbf{L}_{v4} \end{bmatrix} \in \mathbb{R}^{2N \times 4N}, \quad \mathbf{g}_k = \begin{Bmatrix} \mathbf{M}^{-1}\mathbf{f}\,(t_k) \\ \mathbf{M}^{-1}\mathbf{f}\,(t_k + \Delta t/3) \\ \mathbf{M}^{-1}\mathbf{f}\,(t_k + 2\Delta t/3) \\ \mathbf{M}^{-1}\mathbf{f}\,(t_{k+1}) \end{Bmatrix} \in \mathbb{R}^{4N}. \tag{16}$$

Integrals of Eq. (13) can analytically be determined resulting in power series expressions of matrix $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$. After some straight operations, the matrices $\mathbf{T}$ and $\mathbf{L}$ can be written in compact form as

$$\mathbf{T} = \sum_{j=0}^{\infty} \mathbf{t}_j(\Delta t) \otimes \mathbf{A}^j, \quad \mathbf{L} = \sum_{j=0}^{\infty} \boldsymbol{l}_j(\Delta t) \otimes \mathbf{A}^j, \tag{17}$$

where $\otimes$ denotes the Kronecker product and the sequence of Δt-dependent matrices $\{\mathbf{t}_j(\Delta t)\}_{j=0}^{\infty} \in \mathbb{R}^{2\times 2}$ and $\{\boldsymbol{l}_j(\Delta t)\}_{j=0}^{\infty} \in \mathbb{R}^{2\times 4}$, are

$$\mathbf{t}_j(\Delta t) = \frac{(-1)^j\, \Delta t^{2j-1}}{(2j)!} \begin{bmatrix} \Delta t & \Delta t^2/(2j+1) \\ 2j & \Delta t \end{bmatrix}, \quad j \geq 0, \tag{18}$$

$$\boldsymbol{l}_j(\Delta t) = \frac{(-1)^j\, \Delta t^{2j+1}}{(2j+4)!} \begin{bmatrix} \dfrac{(j+1)(8j^2+18j+13)\Delta t}{2j+5} & \dfrac{36(j+1)^2\Delta t}{2j+5} \\ (2j+1)(4j^2+5j+3) & 9(2j+1)^2 \\ -\dfrac{9(2j^2+j-1)\Delta t}{2j+5} & \dfrac{2(1+2j^2)\Delta t}{2j+5} \\ -9(j-1)(2j+1) & 4j^2-4j+3 \end{bmatrix}. \tag{19}$$

The scheme of Eq. (14) corresponds to the initial iteration (undamped) of the asymptotic expansion ($n = 0$). Now it is the turn to obtain the recursive formula that allows to find the rest of the terms of the expansion ($n \geq 1$) for all time samples ($k \geq 0$).

2.2. *Solution of nth iteration, $n \geq 1$*

According to Eq. (7), the nth iteration of the asymptotic expansion, $\mathbf{x}^{(n)}(t)$, is in turn solution of a forced undamped problem where the applied force is proportional to the velocities of the previous iteration. Thus, we can use again the Green's functions-based approach, yielding

$$\begin{aligned} \mathbf{x}_{k+1}^{(n)} &= \mathbf{G}(\Delta t)\, \mathbf{x}_k^{(n)} + \mathbf{H}(\Delta t)\, \dot{\mathbf{x}}_k^{(n)} \\ &\quad - \int_{t=t_k}^{t_{k+1}} \mathbf{H}(t_{k+1}-t)\mathbf{M}^{-1}\mathbf{C}\; \dot{\mathbf{x}}^{(n-1)}(t)\, dt, \\ \dot{\mathbf{x}}_{k+1}^{(n)} &= \dot{\mathbf{G}}(\Delta t)\, \mathbf{x}_k^{(n)} + \dot{\mathbf{H}}(\Delta t)\, \dot{\mathbf{x}}_k^{(n)} \\ &\quad - \int_{t=t_k}^{t_{k+1}} \mathbf{G}(t_{k+1}-t)\mathbf{M}^{-1}\mathbf{C}\; \dot{\mathbf{x}}^{(n-1)}(t)\, dt, \end{aligned} \tag{20}$$

where $\mathbf{x}_k^{(n)} = \mathbf{x}^{(n)}(t_k)$, and $\dot{\mathbf{x}}_k^{(n)} = \dot{\mathbf{x}}^{(n)}(t_k)$, and the initial conditions $\mathbf{x}_0^{(n)} = \dot{\mathbf{x}}_0^{(n)} = \mathbf{0}$. Above, $\dot{\mathbf{x}}^{(n-1)}(t)$ is not explicitly available, since the solution is being determined from the scheme at the samples $t_k,\ k = 0, 1, 2, \ldots$. Equation (20) will be transformed

into an iterative scheme in two stages. First, integration by parts of Eq. (20) in terms of the displacements field $\mathbf{x}^{(n-1)}(t)$ leads to

$$\begin{aligned}
&-\int_{t=t_k}^{t_{k+1}} \mathbf{H}(t_{k+1}-t)\,\mathbf{M}^{-1}\mathbf{C}\;\dot{\mathbf{x}}^{(n-1)}(t)\,dt \\
&\quad = \mathbf{H}(\Delta t)\mathbf{M}^{-1}\mathbf{C}\,\mathbf{x}_k^{(n-1)} \\
&\qquad -\int_{t=t_k}^{t_{k+1}} \mathbf{G}[t_{k+1}-t]\,\mathbf{M}^{-1}\mathbf{C}\;\mathbf{x}^{(n-1)}(t)\,dt, \qquad (21)\\
&-\int_{t=t_k}^{t_{k+1}} \mathbf{G}(t_{k+1}-t)\,\mathbf{M}^{-1}\mathbf{C}\;\dot{\mathbf{x}}^{(n-1)}(t)\,dt \\
&\quad = -\mathbf{G}(0)\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{x}_{k+1}^{(n-1)} + \mathbf{G}(\Delta t)\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{x}_k^{(n-1)} \\
&\qquad + \mathbf{A}\int_{t=t_k}^{t_{k+1}} \mathbf{H}[t_{k+1}-t]\,\mathbf{M}^{-1}\mathbf{C}\;\mathbf{x}^{(n-1)}(t)\,dt, \qquad (22)
\end{aligned}$$

where the relationships between $\mathbf{G}(t)$ and $\mathbf{H}(t)$ and their derivatives given by Eq. (9) have been used. The second stage consist of the approximation of $\mathbf{x}^{(n-1)}(t)$ by cubic splines using both displacements and velocities at $t = t_k$ and $t = t_{k+1}$. Within the interval $[t_k, t_{k+1}]$ the dimensionless variable ξ is defined as

$$\xi = \frac{t-t_k}{t_{k+1}-t_k}, \quad 0 \le \xi \le 1, \quad t_k \le t \le t_{k+1}, \qquad (23)$$

and the value of $\mathbf{x}^{(n-1)}(t)$ is approximated by

$$\mathbf{x}^{(n-1)}(t) \approx \mathcal{N}_1(\xi), \mathbf{x}_k^{(n-1)} + \mathcal{D}_1(\xi)\,\Delta t\,\dot{\mathbf{x}}_k^{(n-1)} + \mathcal{N}_2(\xi)\,\mathbf{x}_{k+1}^{(n-1)} + \mathcal{D}_2(\xi)\,\Delta t\,\dot{\mathbf{x}}_{k+1}^{(n-1)}, \qquad (24)$$

where

$$\begin{aligned}
&\mathcal{N}_1(\xi) = (1-\xi)^2(1+2\xi), \quad \mathcal{N}_2(\xi) = (3-2\xi)\xi^2, \\
&\mathcal{D}_1(\xi) = (1-\xi)^2\xi, \quad \mathcal{D}_2(\xi) = -(1-\xi)\xi^2.
\end{aligned} \qquad (25)$$

By means of this interpolation, the function and its derivative coincide with those evaluated from the approximation at the endpoints of the interval, at $t = \{t_k, t_{k+1}\}$. After plugging Eq. (24) into Eq. (21), the integrals can explicitly be written in terms of the $(n-1)$th iteration at t_k and t_{k+1}, yielding

$$\begin{aligned}
&-\int_{t=t_k}^{t_{k+1}} \mathbf{H}(t_{k+1}-t)\,\mathbf{M}^{-1}\mathbf{C}\;\dot{\mathbf{x}}^{(n-1)}(t)\,dt \\
&\quad = \boldsymbol{\alpha}_{uu}\,\mathbf{x}_k^{(n-1)} + \boldsymbol{\alpha}_{uv}\,\dot{\mathbf{x}}_k^{(n-1)} + \boldsymbol{\beta}_{uu}\,\mathbf{x}_{k+1}^{(n-1)} + \boldsymbol{\beta}_{uv}\,\dot{\mathbf{x}}_{k+1}^{(n-1)}, \\
&-\int_{t=t_k}^{t_{k+1}} \mathbf{G}(t_{k+1}-t)\,\mathbf{M}^{-1}\mathbf{C}\;\dot{\mathbf{x}}^{(n-1)}(t)\,dt \\
&\quad = \boldsymbol{\alpha}_{vu}\,\mathbf{x}_k^{(n-1)} + \boldsymbol{\alpha}_{vv}\,\dot{\mathbf{x}}_k^{(n-1)} + \boldsymbol{\beta}_{vu}\,\mathbf{x}_{k+1}^{(n-1)} + \boldsymbol{\beta}_{vv}\,\dot{\mathbf{x}}_{k+1}^{(n-1)},
\end{aligned} \qquad (26)$$

where the $N \times N$ matrices of coefficients are

$$\begin{aligned}
\boldsymbol{\alpha}_{uu} &= \mathbf{H}(\Delta t)\mathbf{M}^{-1}\mathbf{C} - \Delta t \int_{\xi=0}^{1} \mathbf{G}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{N}_1(\xi)\,d\xi,\\
\boldsymbol{\alpha}_{uv} &= -\Delta t^2 \int_{\xi=0}^{1} \mathbf{G}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{D}_1(\xi)\,d\xi,\\
\boldsymbol{\alpha}_{vu} &= \mathbf{G}(\Delta t)\,\mathbf{M}^{-1}\mathbf{C} + \Delta t\,\mathbf{A} \int_{\xi=0}^{1} \mathbf{H}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{N}_1(\xi)\,d\xi,\\
\boldsymbol{\alpha}_{vv} &= \Delta t^2\,\mathbf{A} \int_{\xi=0}^{1} \mathbf{H}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{D}_1(\xi)\,d\xi,\\
\boldsymbol{\beta}_{uu} &= -\Delta t \int_{\xi=0}^{1} \mathbf{G}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{N}_2(\xi)\,d\xi,\\
\boldsymbol{\beta}_{uv} &= -\Delta t^2 \int_{\xi=0}^{1} \mathbf{G}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{D}_2(\xi)\,d\xi,\\
\boldsymbol{\beta}_{vu} &= -\mathbf{M}^{-1}\mathbf{C} + \Delta t\,\mathbf{A} \int_{\xi=0}^{1} \mathbf{H}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{N}_2(\xi)\,d\xi,\\
\boldsymbol{\beta}_{vv} &= \Delta t^2\,\mathbf{A} \int_{\xi=0}^{1} \mathbf{H}[\Delta t(1-\xi)]\,\mathbf{M}^{-1}\mathbf{C}\,\mathcal{D}_2(\xi)\,d\xi.
\end{aligned} \tag{27}$$

Using the definition of matrix $\mathbf{T}$ from Eq. (15) and the values of the integrals of Eq. (26), the iterative scheme of Eq. (20) can be written in compact form as the following double recursive scheme, both in k and n

$$\mathbf{X}_{k+1}^{(n)} = \mathbf{T}\,\mathbf{X}_{k}^{(n)} + \boldsymbol{\alpha}\,\mathbf{X}_{k}^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_{k+1}^{(n-1)}, \quad n \geq 1, \quad k \geq 0, \tag{28}$$

where the nth state vector and the initial conditions are

$$\mathbf{X}_{k}^{(n)} = \begin{Bmatrix} \mathbf{x}_{k}^{(n)} \\ \dot{\mathbf{x}}_{k}^{(n)} \end{Bmatrix}, \quad \mathbf{X}_{0}^{(n)} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \quad n \geq 1, \tag{29}$$

and the block matrices $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{2N \times 2N}$

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_{uu} & \boldsymbol{\alpha}_{uv} \\ \boldsymbol{\alpha}_{vu} & \boldsymbol{\alpha}_{vv} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_{uu} & \boldsymbol{\beta}_{uv} \\ \boldsymbol{\beta}_{vu} & \boldsymbol{\beta}_{vv} \end{bmatrix}. \tag{30}$$

Integrals of Eq. (27) can analytically be solved using the series expansion of $\mathbf{G}(t)$ and $\mathbf{H}(t)$ given by Eq. (9). After some algebra, the matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be written as

$$\begin{aligned}
\boldsymbol{\alpha} &= \sum_{j=0}^{\infty} \boldsymbol{\alpha}_j(\Delta t) \otimes \left(\mathbf{A}^j\,\mathbf{M}^{-1}\mathbf{C}\right) = \left[\sum_{j=0}^{\infty} \boldsymbol{\alpha}_j(\Delta t) \otimes \mathbf{A}^j\right] \cdot \left(\mathbf{I}_2 \otimes \mathbf{M}^{-1}\mathbf{C}\right),\\
\boldsymbol{\beta} &= \sum_{j=0}^{\infty} \boldsymbol{\beta}_j(\Delta t) \otimes \left(\mathbf{A}^j\,\mathbf{M}^{-1}\mathbf{C}\right) = \left[\sum_{j=0}^{\infty} \boldsymbol{\beta}_j(\Delta t) \otimes \mathbf{A}^j\right] \cdot \left(\mathbf{I}_2 \otimes \mathbf{M}^{-1}\mathbf{C}\right),
\end{aligned} \tag{31}$$

where $\mathbf{I}_2$ denotes the identity matrix of order 2. Above, the sequences $\{\boldsymbol{\alpha}_j(\Delta t)\}$ and $\{\boldsymbol{\beta}_j(\Delta t)\}$, $j \geq 0$, of 2×2 matrices are defined as

$$\boldsymbol{\alpha}_j(\Delta t) = \frac{(-1)^j \Delta t^{2j}}{(2j+4)!} \begin{bmatrix} 12(j+1)\Delta t & -2(2j+1)(j+1)\Delta t^2 \\ 12(2j+1)(j+2) & -4j(2j+1)(j+2)\Delta t \end{bmatrix}, \quad j = 0, 1, 2, \ldots \tag{32}$$

$$\boldsymbol{\beta}_j(\Delta t) = \frac{(-1)^j \Delta t^{2j}}{(2j+4)!} \begin{bmatrix} -12(j+1)\Delta t & 2(2j+1)\Delta t^2 \\ -12(2j+1)(j+2) & 8j(j+2)\Delta t \end{bmatrix}, \quad j = 0, 1, 2, \ldots \tag{33}$$

Equations (14) and (28) provide a well-defined iterative framework that enables the recursive computation in the k- and n-domains of each one of the asymptotic expansion terms, i.e., $\mathbf{X}_k^{(n)}$, starting from $\mathbf{X}_0^{(0)} = \mathbf{U}_0$ and $\mathbf{X}_0^{(n)} = \mathbf{0}$, $n \geq 1$. The benefits of the proposed method emerge after summing the resulting asymptotic series. Let's see it in the following section

3. Proposed Step-by-Step Explicit Scheme

Reaching n iterations, then the transient solution at $t = t_k$ can be approximated by the sequence of partial sums, denoted by $\mathbf{U}_k^{(n)}$. Namely,

$$\mathbf{U}_k^{(n)} = \begin{Bmatrix} \mathbf{u}^{(n)}(t_k) \\ \dot{\mathbf{u}}^{(n)}(t_k) \end{Bmatrix} \equiv \begin{Bmatrix} \mathbf{u}_k^{(n)} \\ \dot{\mathbf{u}}_k^{(n)} \end{Bmatrix} = \mathbf{X}_k^{(0)} + \mathbf{X}_k^{(1)} + \cdots + \mathbf{X}_k^{(n)} = \sum_{\nu=0}^{n} \mathbf{X}_k^{(\nu)}. \tag{34}$$

The limit of the above sequence at $t = t_k$, provided that it exists, represents the proposed approximated solution to the transient problem. It will be denoted by

$$\mathbf{U}_k = \begin{Bmatrix} \mathbf{u}_k \\ \dot{\mathbf{u}}_k \end{Bmatrix} \equiv \lim_{n\to\infty} \mathbf{U}_k^{(n)} = \sum_{n=0}^{\infty} \mathbf{X}_k^{(n)}, \quad k \geq 0. \tag{35}$$

It turns out that the above limit can be evaluated leading to an explicit time integration algorithm. Indeed, writing the $n+1$ first iterations, $0 \leq \nu \leq n$

$$\begin{aligned} \mathbf{X}_{k+1}^{(0)} &= \mathbf{T}\,\mathbf{X}_k^{(0)} + \mathbf{L}\,\mathbf{g}_k \quad \nu = 0, \\ -\boldsymbol{\beta}\,\mathbf{X}_{k+1}^{(0)} + \mathbf{X}_{k+1}^{(1)} &= \mathbf{T}\,\mathbf{X}_k^{(1)} + \boldsymbol{\alpha}\,\mathbf{X}_k^{(0)} \quad \nu = 1, \\ &\vdots \qquad\qquad \vdots \\ -\boldsymbol{\beta}\,\mathbf{X}_{k+1}^{(n-1)} + \mathbf{X}_{k+1}^{(n)} &= \mathbf{T}\,\mathbf{X}_k^{(n)} + \boldsymbol{\alpha}\,\mathbf{X}_k^{(n-1)} \quad \nu = n. \end{aligned} \tag{36}$$

Summing the left and the right side terms of the above equations and taking into account that $\mathbf{U}_k^{(n)} = \sum_{\nu=0}^{n} \mathbf{X}_k^{(\nu)}$, yields

$$-\boldsymbol{\beta}\,\mathbf{U}_{k+1}^{(n-1)} + \mathbf{U}_{k+1}^{(n)} = \mathbf{T}\,\mathbf{U}_k^{(n)} + \boldsymbol{\alpha}\,\mathbf{U}_k^{(n-1)} + \mathbf{L}\mathbf{g}_k, \quad n \geq 1\ . \tag{37}$$

Taking limits now on both sides as $n \to \infty$ and assuming that $\mathbf{U}_k = \lim_{n\to\infty} \mathbf{U}_k^{(n)}$ exists, the following implicit algorithm arises:

$$(\mathbf{I}_{2N} - \boldsymbol{\beta})\ \mathbf{U}_{k+1} = (\mathbf{T} + \boldsymbol{\alpha})\ \mathbf{U}_k + \mathbf{L}\mathbf{g}_k. \tag{38}$$

Multiplying both sides of the equation by $(\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}$, Eq. (38) can be expressed as the explicit scheme

$$\mathbf{U}_{k+1} = \mathbf{a}\,\mathbf{U}_k + \mathbf{b}_k, \quad k = 0, 1, 2, \ldots, \tag{39}$$

where the main matrices of the algorithm, $\mathbf{a} \in \mathbb{R}^{2N\times 2N}$ and $\mathbf{b}_k \in \mathbb{R}^{2N}$, are

$$\mathbf{a} = (\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}\,(\mathbf{T} + \boldsymbol{\alpha}) \in \mathbb{R}^{2N\times 2N}, \tag{40}$$

$$\mathbf{b}_k = (\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}\,\mathbf{L}\,\mathbf{g}_k \in \mathbb{R}^{2N}, \quad k = 0, 1, 2, \ldots \tag{41}$$

The calculation of the inverse matrix does not add any drawback with respect to the loss of accuracy because, as we will see later, the corresponding Neumann series can be invoked. Essentially, the iterative procedure shown in Eq. (39) summarizes the main contribution of this paper. In the following sections, properties of the numerical approach will be discussed, indeed:

- In Sec. 4 and in Appendix A, the conditions for convergence of asymptotic expansion $\mathbf{U}_k = \sum_{n=0}^{\infty} \mathbf{X}_k^{(n)}$ will be proved.
- In Sec. 5, algorithms for computation of the main matrices of the explicit method, say $\mathbf{a}$ and $\mathbf{b}_k$, will be proposed.
- In Sec. 6, the conditions to ensure the stability of the scheme will be investigated.
- In Sec. 7, the accuracy of the approach and its sensitivity with respect to the different parameters are studied in detail

4. Analysis of Convergence

The explicit iterative scheme outlined in Eq. (39) is the result of assuming as true that the limit of Eq. (35) exists, i.e., the series of the asymptotic method $\mathbf{U}_k = \sum_{n=0}^{\infty} \mathbf{X}_k^{(n)}$ is convergent. After some straight derivations (see Appendix A) the terms of the sequence $\{\mathbf{X}_k^{(n)},\ n \geq 0\}$ are linearly dependent on the previous ones through the square matrix $\boldsymbol{\beta}$, shown in Eq. (31). This leads to the following result:

Theorem 1. *The series* $\mathbf{U}_k = \sum_{n=0}^{\infty} \mathbf{X}_k^{(n)}$ *converges provided that* $\rho(\boldsymbol{\beta}) < 1$, *where* $\rho(\bullet)$ *denotes the spectral radius.*

A rigorous proof of this proposition can be found in Appendix A. The matrix $\boldsymbol{\beta}$ has already been presented in Eqs. (31) and (33) as a power series expansion of the system matrix $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$. Moreover, in view of the form of the matrix all terms are proportional to the damping $\mathbf{M}^{-1}\mathbf{C}$. The matrix $\boldsymbol{\beta}$ can also be presented under the form

$$\boldsymbol{\beta} = \Delta t\,\left(\mathbf{I}_2 \otimes \mathbf{M}^{-1}\mathbf{C}\right)\,\left[\lim_{m\to\infty} \boldsymbol{\sigma}_m\left(\Delta t\sqrt{\mathbf{M}^{-1}\mathbf{K}}\right)\right], \tag{42}$$

where the matrix

$$\boldsymbol{\sigma}_m(\boldsymbol{\tau}) = \sum_{j=0}^{m/2} \frac{(-1)^j}{(2j+4)!} \begin{bmatrix} -12(j+1)\,\boldsymbol{\tau}^{2j} & 2(2j+1)\Delta t\,\boldsymbol{\tau}^{2j} \\ -12(2j+1)(j+2)\,\boldsymbol{\tau}^{2j}/\Delta t & 8j(j+2)\,\boldsymbol{\tau}^{2j} \end{bmatrix} \tag{43}$$

has 2×2 elements as long as the argument $\boldsymbol{\tau} \in \mathbb{R}$, but it will be $\boldsymbol{\sigma}(\boldsymbol{\tau}) \in \mathbb{R}^{2N \times 2N}$ if $\boldsymbol{\tau} \in \mathbb{R}^{N \times N}$, as shown in Eq. (42) with $\boldsymbol{\tau} = \Delta t\sqrt{\mathbf{M}^{-1}\mathbf{K}}$. The index m denotes the maximum order of $\boldsymbol{\tau}$ in the power series expansion. For practical cases, a finite value for m means the truncation of the series. In such case, Eq. (42) represents somehow a decomposition of matrix $\boldsymbol{\beta}$ into three terms: (i) a direct dependency of time step Δt, (ii) the matrix $\boldsymbol{\sigma}_m(\Delta t\sqrt{\mathbf{M}^{-1}\mathbf{K}})$ and (iii) the matrix $\mathbf{M}^{-1}\mathbf{C}$. In view of the structure shown in Eq. (42), the higher the damping, the smaller the time step needed to ensure convergence. For real structures with light or moderate damping forces, the convergence of the method is guaranteed for most time-step range used in practice. This point will be discussed in the numerical example.

5. Algorithms for Computation of Main Matrices

To complete the proposed methodology, specific algorithms are needed to calculate the main matrices of the numerical approach, i.e., from Sec. 3, Eq. (39), the matrix $\mathbf{a}(\Delta t)$ and the vector $\mathbf{b}_k(\Delta t)$. Throughout this section, the dependence of the parameter Δt in both matrices will be highlighted. Rewriting both expressions

$$\mathbf{a}(\Delta t) = (\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1} (\mathbf{T} + \boldsymbol{\alpha}), \tag{44}$$

$$\mathbf{b}_k(\Delta t) = (\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1} \mathbf{L}\, \mathbf{g}_k, \quad k = 0, 1, 2, \ldots \tag{45}$$

Both $\mathbf{a}(\Delta t)$ and $\mathbf{b}_k(\Delta t)$ are computed from matrices $\mathbf{T}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\mathbf{L}$, as well as the vector of external forces given by $\mathbf{g}_k$. Apart from the latter, the rest of matrices are series expansions in terms of the time step Δt and of the system matrices, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$, in particular in terms of $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$ and $\mathbf{M}^{-1}\mathbf{C}$. Additionally, the computation of the inverse matrix $(\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}$ is also required. The condition for the convergence of the method $\rho(\boldsymbol{\beta}) < 1$ (see Theorem 1), enables now to use the Neumann series for its computation by means of the formula

$$(\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1} = \sum_{r=0}^{\infty} \boldsymbol{\beta}^r \approx \mathbf{I}_{2N} + \boldsymbol{\beta} + \boldsymbol{\beta}^2 + \cdots + \boldsymbol{\beta}^r. \tag{46}$$

In the following points the algorithms for the efficient computation of both $\mathbf{a}(\Delta t)$ and $\mathbf{b}_k(\Delta t)$ will be described in detail. Although the same matrix $\boldsymbol{\beta}$ arises in both Eqs. (44) and (45), it will be computed differently in both cases, using different time steps and truncation orders in the series involved.

5.1. *Precise computation of matrix* $\mathbf{a}(\Delta t)$

Accuracy in the evaluation of the main matrix of the algorithm $\mathbf{a}(\Delta t)$ is a key issue to achieve satisfactory numerical results. In the absence of external forces it yields

$$\mathbf{U}_{k+1} = \mathbf{a}(\Delta t)\, \mathbf{U}_k. \tag{47}$$

Therefore, it is verified then that $\mathbf{a}(\Delta t) = \mathbf{a}(\Delta t/2)\mathbf{a}(\Delta t/2) = [\mathbf{a}(\Delta t/4)]^4$. Consider a positive integer p and the reduced time step defined as $\Delta t_0 = \Delta t/2^p$ then

$$\mathbf{a}(\Delta t) = \left[\mathbf{a}\left(\frac{\Delta t}{2^p}\right)\right]^{2^p} = [\mathbf{a}(\Delta t_0)]^{2^p}. \tag{48}$$

Moreover, $\mathbf{a}(\Delta t_0)$ can be expressed in incremental form as

$$\mathbf{a}(\Delta t_0) = \mathbf{I}_{2N} + \delta\mathbf{a}(\Delta t_0), \tag{49}$$

where $\delta\mathbf{a}(\Delta t_0)$ is a matrix whose values are very small. In fact, it can be stated that $\lim_{\Delta t\to 0} \delta\mathbf{a}(\Delta t) = \mathbf{0}$. The relationship of Eq. (48) enables the use of the so-called 2^p algorithm, commonly used for the evaluation of exponential matrices [Moler and Loan (2003)]. Moreover, from Eq. (49), this technique can be used on the incremental part $\delta\mathbf{a}(\Delta t)$ instead of on the total matrix, improving significantly the accuracy. Thus, the product $\mathbf{a}(\Delta t/2)\mathbf{a}(\Delta t/2) = \mathbf{a}(\Delta t) = \mathbf{I}_{2N} + \delta\mathbf{a}(\Delta t)$ produces

$$\delta\mathbf{a}(\Delta t) = 2\,\delta\mathbf{a}(\Delta t/2) + \delta\mathbf{a}(\Delta t/2)\cdot\delta\mathbf{a}(\Delta t/2). \tag{50}$$

Let us consider the reduced time-step introduced above, i.e., $\Delta t_0 = \Delta t/2^p$. The first step is to compute $\delta\mathbf{a}(\Delta t_0)$. As pointed out before, this latter is very small in the sense that $\|\delta\mathbf{a}(\Delta t_0)\| \ll 1$ for any matrix norm. Therefore, the iterative process of Eq. (50) results much more effective from a round-off error perspective than evaluating the total matrices $\mathbf{a}(\Delta t)$. The iterative process consists of p steps, namely

$$\begin{aligned}
\delta\mathbf{a}(\Delta t/2^{p-1}) &= 2\,\delta\mathbf{a}(\Delta t/2^p) + \delta\mathbf{a}(\Delta t/2^p)\cdot\delta\mathbf{a}(\Delta t/2^p),\\
\delta\mathbf{a}(\Delta t/2^{p-2}) &= 2\,\delta\mathbf{a}(\Delta t/2^{p-1}) + \delta\mathbf{a}(\Delta t/2^{p-1})\cdot\delta\mathbf{a}(\Delta t/2^{p-1}),\\
&\vdots \quad (p \text{ times}) \quad \vdots\\
\delta\mathbf{a}(\Delta t) &= 2\,\delta\mathbf{a}(\Delta t/2) + \delta\mathbf{a}(\Delta t/2)\cdot\delta\mathbf{a}(\Delta t/2),
\end{aligned} \tag{51}$$

and finally the main matrix of the algorithm is determined as $\mathbf{a}(\Delta t) = \mathbf{I}_{2N} + \delta\mathbf{a}(\Delta t)$. For practical applications, it is sufficient to take $p = 20$ to obtain results with the required accuracy.

In order to complete the procedure, it remains to describe how to compute the incremental matrix $\delta\mathbf{a}(\Delta t_0) = \mathbf{a}(\Delta t_0) - \mathbf{I}_{2N}$. As Eq. (44) shows, the computation of $\mathbf{a}(\Delta t_0)$ requires the truncation of two series: (i) those of $\mathbf{T}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and (ii) the Neumann series corresponding to $(\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}$. This lead us to introduce the following two parameters

(i) The parameter m_a stands for the truncation order of matrices $\mathbf{T}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$. For the sake of convenience, we will use a common criterion, that is, fixed the parameter m_a the three series will have the same number of terms. In order to avoid confusion between the exact values and those approximated by truncation, the subscript $(\bullet)_a$ will be used for the latter. Thus, consider any even number

$m_a \geq 2$ and let us introduce following approximations in terms o the reduced time-step Δt_0

$$\mathbf{T} \approx \mathbf{I}_{2N} + \delta\mathbf{T}_a(\Delta t_0) = \mathbf{I}_{2N} + \begin{bmatrix} \delta\mathbf{G}_a(\Delta t_0) & \mathbf{H}_a(\Delta t_0) \\ -\mathbf{A}\,\mathbf{H}_a(\Delta t_0) & \delta\mathbf{G}_a(\Delta t_0) \end{bmatrix},$$

$$\delta\mathbf{G}_a(\Delta t_0) = \sum_{j=1}^{m_a/2} \frac{(-1)^j}{(2j)!} \left(\Delta t_0^2 \mathbf{A}\right)^j, \tag{52}$$

$$\mathbf{H}_a(\Delta t_0) = \Delta t_0 \sum_{j=0}^{m_a/2} \frac{(-1)^j}{(2j+1)!} \left(\Delta t_0^2 \mathbf{A}\right)^j,$$

$$\boldsymbol{\alpha} \approx \boldsymbol{\alpha}_a = \sum_{j=0}^{m_a/2} \boldsymbol{\alpha}_j(\Delta t_0) \otimes \left(\mathbf{A}^j\,\mathbf{M}^{-1}\mathbf{C}\right) = \left[\sum_{j=0}^{m_a/2} \boldsymbol{\alpha}_j(\Delta t_0) \otimes \mathbf{A}^j\right] \cdot \left(\mathbf{I}_2 \otimes \mathbf{M}^{-1}\mathbf{C}\right), \tag{53}$$

$$\boldsymbol{\beta} \approx \boldsymbol{\beta}_a = \sum_{j=0}^{m_a/2} \boldsymbol{\beta}_j(\Delta t_0) \otimes \left(\mathbf{A}^j\,\mathbf{M}^{-1}\mathbf{C}\right) = \left[\sum_{j=0}^{m_a/2} \boldsymbol{\beta}_j(\Delta t_0) \otimes \mathbf{A}^j\right] \cdot \left(\mathbf{I}_2 \otimes \mathbf{M}^{-1}\mathbf{C}\right),$$

where $\boldsymbol{\alpha}_j(\Delta t_0), \boldsymbol{\beta}_j(\Delta t_0) \in \mathbb{R}^{2\times 2}$ are defined in Eqs. (32) and (33) for $j \geq 0$. Since the order of Δt_0 grows by two, a truncation up to $j = m_a/2$ means that the order of Δt_0 will be at least m_a.

(ii) The other parameter needed to compute $\delta\mathbf{a}(\Delta t_0)$ will be designed by r_a and it represents the order of truncation of the Neumann series for $(\mathbf{I}_{2N} - \boldsymbol{\beta}_a)^{-1}$, yielding

$$\left(\mathbf{I}_{2N} - \boldsymbol{\beta}_a\right)^{-1} \approx \mathbf{I}_{2N} + \boldsymbol{\beta}_a + \boldsymbol{\beta}_a^2 + \cdots + \boldsymbol{\beta}_a^{r_a} \equiv \mathbf{I}_{2N} + \delta\boldsymbol{\beta}_a, \tag{54}$$

where the matrix $\delta\boldsymbol{\beta}_a = \boldsymbol{\beta}_a + \boldsymbol{\beta}_a^2 + \cdots + \boldsymbol{\beta}_a^{r_a}$

Substituting Eqs. (52), (53) and (54) into Eq. (44), and after some matrix products we have

$$\begin{aligned} \mathbf{a}(\Delta t_0) &= \left(\mathbf{I}_{2N} - \boldsymbol{\beta}\right)^{-1} \left(\mathbf{T} + \boldsymbol{\alpha}\right) \approx \left(\mathbf{I}_{2N} + \delta\boldsymbol{\beta}_a\right) \left(\mathbf{I}_{2N} + \delta\mathbf{T}_a + \boldsymbol{\alpha}_a\right) \\ &= \mathbf{I}_{2N} + \left(\delta\mathbf{T}_a + \boldsymbol{\alpha}_a + \delta\boldsymbol{\beta}_a + \delta\boldsymbol{\beta}_a\,\delta\mathbf{T}_a + \delta\boldsymbol{\beta}_a\,\boldsymbol{\alpha}_a\right) \equiv \mathbf{I}_{2N} + \delta\mathbf{a}(\Delta t_0), \end{aligned} \tag{55}$$

where consequently the incremental part $\delta\mathbf{a}(\Delta t_0)$ is

$$\delta\mathbf{a}(\Delta t_0) = \delta\mathbf{T}_a + \boldsymbol{\alpha}_a + \delta\boldsymbol{\beta}_a + \delta\boldsymbol{\beta}_a\,\delta\mathbf{T}_a + \delta\boldsymbol{\beta}_a\,\boldsymbol{\alpha}_a, \tag{56}$$

Taking $p = 20$, then $\Delta t_0 = \Delta t/2^{20} = \Delta t/1\,048\,576$ is a very small quantity so that it is sufficient to take $m_a = 2$ to achieve a highly precise estimation of $\delta\mathbf{a}(\Delta t_0)$. In

addition, the case $m_a = 2$ presents advantages with respect to the stability, as it will be discussed later in Sec. 6. The form of the matrices is transcribed below for this particular case ($m_a = 2$).

$$\delta\mathbf{T}_a = \begin{bmatrix} -\frac{\Delta t_0^2}{2}\mathbf{A} & \Delta t_0\mathbf{I}_N - \frac{\Delta t_0^3}{6}\mathbf{A} \\ -\Delta t_0\mathbf{A} + \frac{\Delta t_0^3}{6}\mathbf{A}^2 & -\frac{\Delta t_0^2}{2}\mathbf{A} \end{bmatrix}, \tag{57}$$

$$\boldsymbol{\alpha}_a = \begin{bmatrix} \frac{\Delta t_0}{2}\mathbf{M}^{-1}\mathbf{C} - \frac{\Delta t_0^3}{30}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} & -\frac{\Delta t_0^2}{12}\mathbf{M}^{-1}\mathbf{C} + \frac{\Delta t_0^4}{60}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} \\ \mathbf{M}^{-1}\mathbf{C} - \frac{3\Delta t_0^2}{20}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} & \frac{\Delta t_0^3}{20}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} \end{bmatrix}, \tag{58}$$

$$\boldsymbol{\beta}_a = \begin{bmatrix} -\frac{\Delta t_0}{2}\mathbf{M}^{-1}\mathbf{C} + \frac{\Delta t_0^3}{30}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} & \frac{\Delta t_0^2}{12}\mathbf{M}^{-1}\mathbf{C} - \frac{\Delta t_0^4}{120}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} \\ -\mathbf{M}^{-1}\mathbf{C} + \frac{3\Delta t_0^2}{20}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} & -\frac{\Delta t_0^3}{30}\mathbf{A}\mathbf{M}^{-1}\mathbf{C} \end{bmatrix}, \tag{59}$$

$$\delta\boldsymbol{\beta}_a = \boldsymbol{\beta}_a + \boldsymbol{\beta}_a^2 + \cdots + \boldsymbol{\beta}_a^{r_a}. \tag{60}$$

Algorithm 1. outlines the steps necessary to obtain the matrix $\mathbf{a}(\Delta t)$. Since in general $\Delta t_0 \ll T$, the two parameters m_a and r_a will be taken as $m_a = r_a = 2$. As will be shown later, the scheme is conditionally stable for these values for systems with nonzero damping.

5.2. *Computation of vector* $\mathbf{b}_k(\Delta t)$

The vector $\mathbf{b}_k(\Delta t) = (\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}\,\mathbf{L}\,\mathbf{g}_k$ is directly related to the nonhomogeneous terms of the transient problem. On one hand, it depends on the inverse matrix $(\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}$ and, on the other hand, on the matrix $\mathbf{L}$ defined above in Eqs. (17)

Algorithm 1. Computation of main matrix $\mathbf{a}(\Delta t)$

1: Fix main parameters, Δt, $r_a = 2$, $m_a = 2$
2: Compute matrices $\mathbf{M}^{-1}\mathbf{C}$, $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$
3: Evaluate $\Delta t_0 = \Delta t/2^p$ $(p = 20)$
4: Compute matrix $\delta\mathbf{T}_a$, Eq. (57)
5: Compute matrix $\boldsymbol{\alpha}_a$, Eq. (58)
6: Compute matrix $\boldsymbol{\beta}_a$, Eq. (59)
7: Compute matrix $\delta\boldsymbol{\beta}_a = \boldsymbol{\beta}_a + \boldsymbol{\beta}_a^2$, Eq. (60)
8: Compute matrix $\delta\mathbf{a}(\Delta t_0) = \delta\mathbf{T}_a + \boldsymbol{\alpha}_a + \delta\boldsymbol{\beta}_a + \delta\boldsymbol{\beta}_a\,\delta\mathbf{T}_a + \delta\boldsymbol{\beta}_a\,\boldsymbol{\alpha}_a$, Eq. (56)
9: Initialize $\delta\mathbf{a} := \delta\mathbf{a}(\Delta t_0)$
10: **for** $j = 1 \ldots q$ **do**
11: $\quad\delta\mathbf{a} = 2\delta\mathbf{a} + \delta\mathbf{a}\cdot\delta\mathbf{a}$, Eq . (51)
12: **end for**
13: Update $\delta\mathbf{a}(\Delta t) = \delta\mathbf{a}$
14: Evaluate $\mathbf{a}(\Delta t) = \mathbf{I}_{2N} + \delta\mathbf{a}(\Delta t)$

and (19). Both $\boldsymbol{\beta}$ and $\mathbf{L}$ are defined in terms of matrix series and consequently they must be truncated for numerical practice. For that, let us introduce again two parameters:

(i) m_b stands for the order of truncation of the matrix series $\boldsymbol{\beta}$ and $\mathbf{L}$. Thus, the vector $\mathbf{b}_k$ results from the following approximation

$$\mathbf{b}_k = (\mathbf{I}_{2N} - \boldsymbol{\beta})^{-1}\,\mathbf{L}\,\mathbf{g}_k \approx (\mathbf{I}_{2N} - \boldsymbol{\beta}_b)^{-1}\,\mathbf{L}_b\,\mathbf{g}_k, \quad k = 0, 1, 2, \ldots, \tag{61}$$

where

$$\mathbf{L}_b = \sum_{j=0}^{m_b/2} \boldsymbol{l}_j(\Delta t) \otimes \mathbf{A}^j, \quad \boldsymbol{\beta}_b = \sum_{j=0}^{m_b/2} \boldsymbol{\beta}_j(\Delta t) \otimes \left(\mathbf{A}^j\,\mathbf{M}^{-1}\mathbf{C}\right), \tag{62}$$

with $\boldsymbol{l}_j(\Delta t)$ and $\boldsymbol{\beta}_j(\Delta t)$, 2×2 matrices defined in Eqs. (19) and (33), respectively. It is important to highlight that both $\mathbf{L}_b$ and $\boldsymbol{\beta}_b$ are computed using the time step of the problem Δt and not the reduced one, Δt_0. As before, the parameter m_b should be an even number because $m_b/2$ denotes the number of terms to be taken.

(ii) r_b denotes the order of truncation of the Neumann series associated to the evaluation of $(\mathbf{I}_{2N} - \boldsymbol{\beta}_b)^{-1} \approx \mathbf{I}_{2N} + \boldsymbol{\beta}_b + \boldsymbol{\beta}_b^2 + \cdots + \boldsymbol{\beta}_b^{r_b}$. Hence, the final evaluation of $\mathbf{b}_k$ leads to

$$\mathbf{b}_k \approx \left(\sum_{n=0}^{r_b} \boldsymbol{\beta}_b^n\right) \mathbf{L}_b\,\mathbf{g}_k. \tag{63}$$

In order to optimize the number of matrix products, the above Neumann series can be computed using the algorithm

$$\begin{aligned} &\mathbf{I}_{2N} + \boldsymbol{\beta} + \boldsymbol{\beta}^2 + \cdots + \boldsymbol{\beta}^r + \cdots \\ &\quad = \mathbf{I}_{2N} + \boldsymbol{\beta} + \boldsymbol{\beta}^2(\mathbf{I}_{2N} + \boldsymbol{\beta} + \boldsymbol{\beta}^2(\mathbf{I}_{2N} + \boldsymbol{\beta} + \cdots)), \end{aligned} \tag{64}$$

requiring consequently just $r_b/2$ matrix products. To ensure that the above sum converges, we know that the spectral radius of $\boldsymbol{\beta}_b$ must verify $\rho(\boldsymbol{\beta}_b) < 1$. In general, increasing the parameter m_b will increase the accuracy of the results and decrease the spectral radius of $\boldsymbol{\beta}_b$, however it will increase the computational cost.

In the Algorithm 2. the necessary steps to obtain the matrix $\mathbf{b}_k(\Delta t)$ are listed. As shown by the numerical experiments, the parameters m_b and r_b result to be significant for the accuracy in certain cases, hence no particular value are recommended here. Later in the numerical examples, a study of the sensitivity of m_b and r_b will be carried out.

Algorithm 2. Computation of main matrix $\mathbf{b}_k(\Delta t)$

Fix main parameters, Δt and r_b, m_b (even numbers)
Compute matrices $\mathbf{M}^{-1}\mathbf{C}$, $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$
Compute and store matrices $\mathbf{A}^j,\ j = 2, \ldots, m_b/2$
Compute and store matrices $\mathbf{A}^j(\mathbf{M}^{-1}\mathbf{C}),\ j = 2, \ldots, m_b/2$
Compute sequence of matrices $\boldsymbol{l}_j(\Delta t) \in \mathbb{R}^{2\times 4},\ 1 \leq j \leq m_b/2,$ Eq. (19)
Compute sequence of matrices $\boldsymbol{\beta}_j(\Delta t) \in \mathbb{R}^{2\times 2},\ 1 \leq j \leq m_b/2,$ Eq. (33)
Compute matrix $\mathbf{L}_b,$ Eq. (62)
Compute matrix $\boldsymbol{\beta}_b,$ Eq. (62)
Compute matrix $\mathbf{B} = \mathbf{I}_{2N} + \boldsymbol{\beta}_b + \boldsymbol{\beta}_b^2$
if $r_b \geq 4$ **then**
    for $n = 2 \ldots r_b/2$ **do**
        $\mathbf{B} = \mathbf{I}_{2N} + \boldsymbol{\beta}_b + \boldsymbol{\beta}_b^2\,\mathbf{B}$
    end for
end if
Compute vector $\mathbf{g}_k,$ Eq. (16)
Compute $\mathbf{b}_k(\Delta t) = \mathbf{B}\,\mathbf{L}_b\,\mathbf{g}_k,$ Eq. (63)

6. Algorithm Stability

The proposed explicit scheme

$$\mathbf{U}_{k+1} = \mathbf{a}(\Delta t)\,\mathbf{U}_k + \mathbf{b}_k(\Delta t), \quad k = 0, 1, 2, \ldots \tag{65}$$

is stable provided that every eigenvalue of the main matrix $\mathbf{a}(\Delta t)$ is less than unity in absolute value, something that is ensured provided that

$$\rho\,[\mathbf{a}(\Delta t)] \leq 1. \tag{66}$$

To discuss the stability of the method it is sufficient to study the single degree-of-freedom oscillator of mass M, spring rigidity K and damping coefficient $c = 2M\omega\,\zeta$. Where $\omega = \sqrt{K/M} = 2\pi/T$ stands for the undamped natural frequency, T for the natural period and ζ is the damping ratio. The matrices of the system are then the numbers $\mathbf{A} = [K/M] = \left[\omega^2\right]$ and $\mathbf{M}^{-1}\mathbf{C} = [2\omega\zeta]$. Since the matrix $\mathbf{a}(\Delta t)$ is computed using the 2^p algorithm, the stability should be studied when it is evaluated at the reduced time step $\Delta t_0 = \Delta t/2^p$. As described above, the order of truncation of the series involved is controlled by both the parameters m_a and r_a. The parameter m_a denotes the truncation order of matrices $\mathbf{T}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ while r_a stands for that of the Neumann series for $(\mathbf{I} - \boldsymbol{\beta})^{-1}$. We can then write

$$\mathbf{a}(\Delta t_0) = (\mathbf{I}_2 - \boldsymbol{\beta})^{-1}(\mathbf{T} + \boldsymbol{\alpha}) \approx \left(\sum_{n=0}^{r_a} \boldsymbol{\beta}_a^n\right)(\mathbf{T}_a + \boldsymbol{\alpha}_a), \tag{67}$$

where the truncated matrices, denoted with the subscript $(\bullet)_a$, results

$$\mathbf{T}_a = \sum_{j=0}^{m_a/2} \frac{(-1)^j \tau^{2j}}{(2j+1)!} \begin{bmatrix} 2j+1 & \Delta t_0 \\ -\tau^2/\Delta t_0 & 2j+1 \end{bmatrix},$$

$$\boldsymbol{\alpha}_a = \sum_{j=0}^{m_a/2} \frac{(-1)^j \tau^{2j+1} \zeta}{(2j+4)!} \begin{bmatrix} 24(j+1) & -4(2j+1)(j+1)\Delta t_0 \\ 24(2j+1)(j+2)/\Delta t_0 & -8(2j+1)(j+2) \end{bmatrix},$$

$$\boldsymbol{\beta}_a = \sum_{j=0}^{m_a/2} \frac{(-1)^j \tau^{2j+1} \zeta}{(2j+4)!} \begin{bmatrix} -24(j+1) & 4(2j+1)\Delta t_0 \\ -24(2j+1)(j+2)/\Delta t_0 & 16j(j+2) \end{bmatrix}. \tag{68}$$

The variable $\tau = \Delta t_0 \omega = 2\pi \Delta t_0 / T$ represents a dimensionless measure of the reduced time step. Eigenvalues λ of matrix $\mathbf{a}(\Delta t_0)$ can be determined as function of the nondimensional time step $\Delta t_0/T = \tau/2\pi$. As shown in Eq. (68), $|\lambda|$ depends also on the parameters m_a, r_a and on the damping ratio ζ. Since the reduced time step necessary to compute $\mathbf{a}(\Delta t_0)$ is a very small number, i.e., $\Delta t_0 = \Delta t/2^p$, with $p = 20$, then in general the inequality $\rho(\boldsymbol{\beta}_a) \ll 1$ holds and only a few terms in the Neumann series are necessary. In particular, assuming the lowest value for r_a,

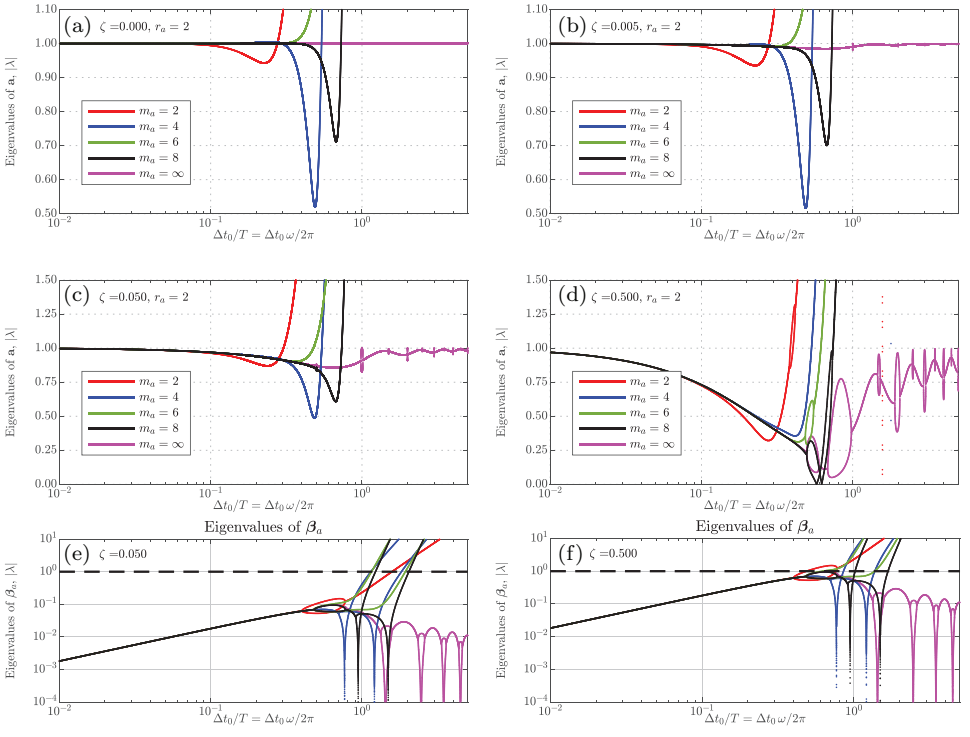

Fig. 1. (Color online) Absolute value of eigenvalues of matrix $\mathbf{a}(\Delta t_0) = (\mathbf{I}_2 - \boldsymbol{\beta})^{-1}(\mathbf{T} + \boldsymbol{\alpha})$ in a single dof system with natural frequency $\omega = \sqrt{K/M} = 2\pi/T$ and damping ratio $\zeta = c/2M\omega$, for different cases of damping level: (a) $\zeta = 0$, (b) $\zeta = 0.005$, (c) $\zeta = 0.05$ and (d) $\zeta = 0.50$. Abscissas represent the reduced time step $\Delta t_0/T$. Curves for different truncation orders m_a are plotted in colors: $m_a = 2$ (red), $m_a = 4$ (blue), $m_a = 6$ (green), $m_a = 8$ (black), $m_a = \infty$ (magenta). The inverse matrix is approximated by $(\mathbf{I} - \boldsymbol{\beta})^{-1} \approx \mathbf{I} + \boldsymbol{\beta}_a + \boldsymbol{\beta}_a^2$ ($r_a = 2$). Plots (e) and (f): absolute value of eigenvalues of matrix $\boldsymbol{\beta}_a$ for two damping cases: (e) $\zeta = 0.05$, (f) $\zeta = 0.50$.

i.e., $r_a = 2$, it leads to very accurate results, something that is supported by the numerical experiments.

Figures 1(a)–1(d) show the absolute value of eigenvalues of matrix $\mathbf{a}(\Delta t_0)$ corresponding to four cases of damping intensity, from $\zeta = 0$ (undamped case) to $\zeta = 0.5$, and for several truncation orders from $m_a = 2$ to $m_a = 8$. Intervals of stability, found as intersections of curves with $|\lambda| = 1$, are listed in Table 1 in terms of $\Delta t_0/T$. For the undamped case ($\zeta = 0$), the orders of truncation for $m_a = 4$ and $m_a = 8$ are unstable for small values of Δt_0. However the case $m_a = 2$ results to be stable for the undamped case in the range $0 < \Delta t_0/T < 0.2757$. Therefore, taking $p = 20$, the stability extends up to the limits $0 < \Delta t/T < 2.89 \times 10^5$, much wider than the usual range of Δt used to guarantee a minimum of accuracy in real problems. As Fig. 1 show, dissipative forces are in general favorable for stability in the proposed scheme. Most conflictive cases in terms of stability are those with null or very light damping, as for example $m_a = 4$ and $m_a = 8$ (see Table 1). The case $m_a = 2$ behaves very favorably, showing conditional stability in an interval that increases slightly as the damping increases.

In Figs. 1(e) and 1(f), the absolute value of eigenvalues of matrix $\boldsymbol{\beta}_a$ are plotted for two cases of damping $\zeta = 0.05$ and $\zeta = 0.5$. Since $\boldsymbol{\beta}_a$ is directly proportional to ζ, the spectral radius decreases also proportionally to the damping level. Convergence interval is the range of $\Delta t_0/T$ within which the inequality $\rho(\boldsymbol{\beta}_a) < 1$ holds. Hence, highly damped structures could have an interval of convergence narrower than the interval of stability, in terms of $\Delta t_0/T$. This is not relevant in practice because of the very small size of $\Delta t_0/T$.

Table 1. Intervals of stability in terms of $\Delta t_0/T$, found as intersections of curves with $|\lambda| = 1$. The reduced time step is $\Delta t_0 = \Delta t/2^p$, with $p = 20$. ζ stands for the damping ratio of the single dof oscillator of frequency $\omega = 2\pi/T$.

Damping ratio	Truncation order	Stability boundaries
$\zeta = 0.000$	$m_a = 2$	$0.0000 < \Delta t_0/T < 0.2757$
	$m_a = 4$	$0.2964 < \Delta t_0/T < 0.5405$
	$m_a = 6$	$0.0000 < \Delta t_0/T < 0.2808$
	$m_a = 8$	$0.2749 < \Delta t_0/T < 0.7279$
$\zeta = 0.005$	$m_a = 2$	$0.0000 < \Delta t_0/T < 0.2791$
	$m_a = 4$	$0.0000 < \Delta t_0/T < 0.5406$
	$m_a = 6$	$0.0000 < \Delta t_0/T < 0.3741$
	$m_a = 8$	$0.0000 < \Delta t_0/T < 0.7287$
$\zeta = 0.050$	$m_a = 2$	$0.0000 < \Delta t_0/T < 0.3024$
	$m_a = 4$	$0.0000 < \Delta t_0/T < 0.5421$
	$m_a = 6$	$0.0000 < \Delta t_0/T < 0.4766$
	$m_a = 8$	$0.0000 < \Delta t_0/T < 0.7343$
$\zeta = 0.500$	$m_a = 2$	$0.0000 < \Delta t_0/T < 0.3871$
	$m_a = 4$	$0.0000 < \Delta t_0/T < 0.5342$
	$m_a = 6$	$0.0000 < \Delta t_0/T < 0.6156$
	$m_a = 8$	$0.0000 < \Delta t_0/T < 0.7407$

7. Numerical Example

Consider the 12-dof discrete lumped spring-mass system shown in Fig. 2 (top). Mass, rigidity and damping coefficient are, respectively, $M = 1$ kg, $K = 100$ N/m and $C = 2\sqrt{KM}\zeta$, where ζ is a dimensionless parameter used to quantify the level of damping in the model. The value $\zeta = 0$ leads to the undamped state with natural frequencies given in Table 2. As ζ increases, the coefficients of the damping matrix also increase linearly. Table 2 also lists the relationships between the modal damping ratios and the parameter ζ, i.e., ξ_j/ζ, where $\xi_j = C'_{jj}/2\omega_j$ and C'_{jj} denotes the jth main diagonal entree of the modal damping matrix, $\mathbf{C}' = \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi}$. The minimum period of the structure is $T = 0.32$ s. Although the damping model is nonproportional, i.e., $C'_{jk} \neq 0,\ j \neq k$, the modal damping ratios represent a good indicator to measure the dissipative level. The external force $f(t)$ is located at the third dof and it reproduces a multiharmonic input affected by a Gaussian weight. The mathematical model and the corresponding graph are shown in Fig. 2 (bottom)

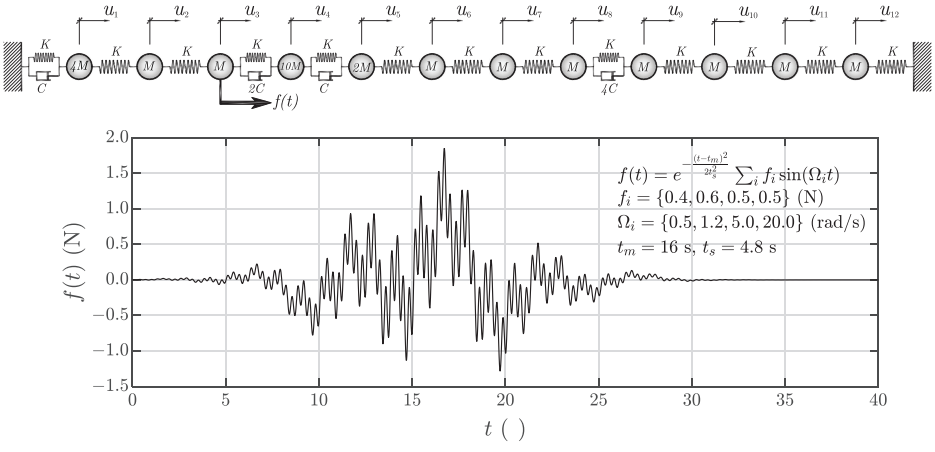

Fig. 2. Discrete system of Example 1. (Top) 12-dof sketch of the lumped-mass structure. External applied force is located at dof # 3. (Bottom) Time function of external force $f(t)$, formed by a superposition of harmonic functions weighted by a Gaussian function.

Table 2. Example 1. Natural frequencies ω_j and modal damping ratios $\xi_j = C'_{jj}/2\omega_j$ (relative to the damping parameter ζ).

Mode, j	1	2	3	4	5	6
Natural frequency, ω_j (rad/s)	1.5482	3.7604	5.4508	6.7428	9.2233	11.0287
Modal damping ratio, $C'_{jj}/2\omega_j\zeta$	0.0972	0.0979	0.5608	0.7106	0.3382	1.0004

Mode j	7	8	9	10	11	12
ω_j (rad/s)	11.8809	14.5269	16.8189	17.5123	18.5560	19.6346
$C'_{jj}/2\omega_j\zeta$	0.8391	0.3872	1.3389	0.5817	0.1603	1.8695

and it is formed by four harmonic functions with different amplitudes. The proposed method will be compared with some existing iterative numerical methods (listed below). Three implicit schemes (Newmark, Wilson and Bathe) and three explicit schemes (Modified Precise Integration Method, Runge–Kutta and the proposed one) have been chosen for the comparison. The time step Δt is a common parameter for all the methods.

Newmark method. The Newmark-β integration scheme [Bathe (2014); Newmark (1959)] is based on the application of the extended mean value theorem for the estimation of displacements and velocities, $\mathbf{u}_{k+1}, \dot{\mathbf{u}}_{k+1}$, from accelerations. For that, two parameters γ and β are used in the method, which will be taken in our particular examples as $\gamma = 1/2$ and $\beta = 1/4$, guaranteeing the stability.

Wilson method. The Wilson-θ method [Bathe (2014); Wilson *et al.* (1973)] assumes a linear change of acceleration within the time range $[t, t + \theta\Delta t]$. The method is implemented in the current numerical examples with $\theta = 1.40$ which ensures the stability of the system [Wilson *et al.* (1973)].

Bathe method. Bathe's method [Bathe (2007); Bathe and Baig (2005); Bathe and Noh (2012)] is an implicit scheme which consists of splitting up the time-step into two sub-steps of length $\gamma\Delta t$ and $(1-\gamma)\Delta t$, respectively, where γ is a parameter. Two different approaches are applied within each sub-step: the trapezoidal rule is used for the first sub-step and the three-point Euler backward method for the second one. For the current numerical examples, the case $\gamma = 1/2$ will be considered, which makes both time sub-steps of equal length, exhibiting in addition unconditional stability.

Modified Precise Integration Method (MPIM). Wan-Xie (2004); Wang and Zhou (2005); Zhong and Williams (1994) The Modified Precise Integration Method combines two techniques: on one hand, it uses the original approach of Zhong Zhong and Williams (1994) for calculating the exponential matrix in the homogeneous solution based on the 2^p algorithm. On the other hand, the integrals of the non-homogeneous terms are accomplished using the Gauss quadrature method with g integration points [Wang and Zhou (2005)]. In Ref. Wang and Zhou (2005), simulations are carried out for a number of integration points equal to $g = \{3, 4, 5\}$. So, in the current paper the same number will be used for comparison.

Runge–Kutta (RK4). The classic fourth-order Runge–Kutta method (RK4, [Burden and Faires (2001); Butcher (2008)]) is not widely used in structural dynamics, but we have found it interesting to include it as example of an explicit method, with good properties of accuracy but conditionally stable.

Proposed method. As described in the theoretical developments, several parameters can be tunned in the proposed method. Along the numerical examples the main matrix $\mathbf{a}(\Delta t)$ is computed using $m_a = r_a = 2$ since highly accurate results are achieved for these values, guaranteeing stability for a very wide range of time steps.

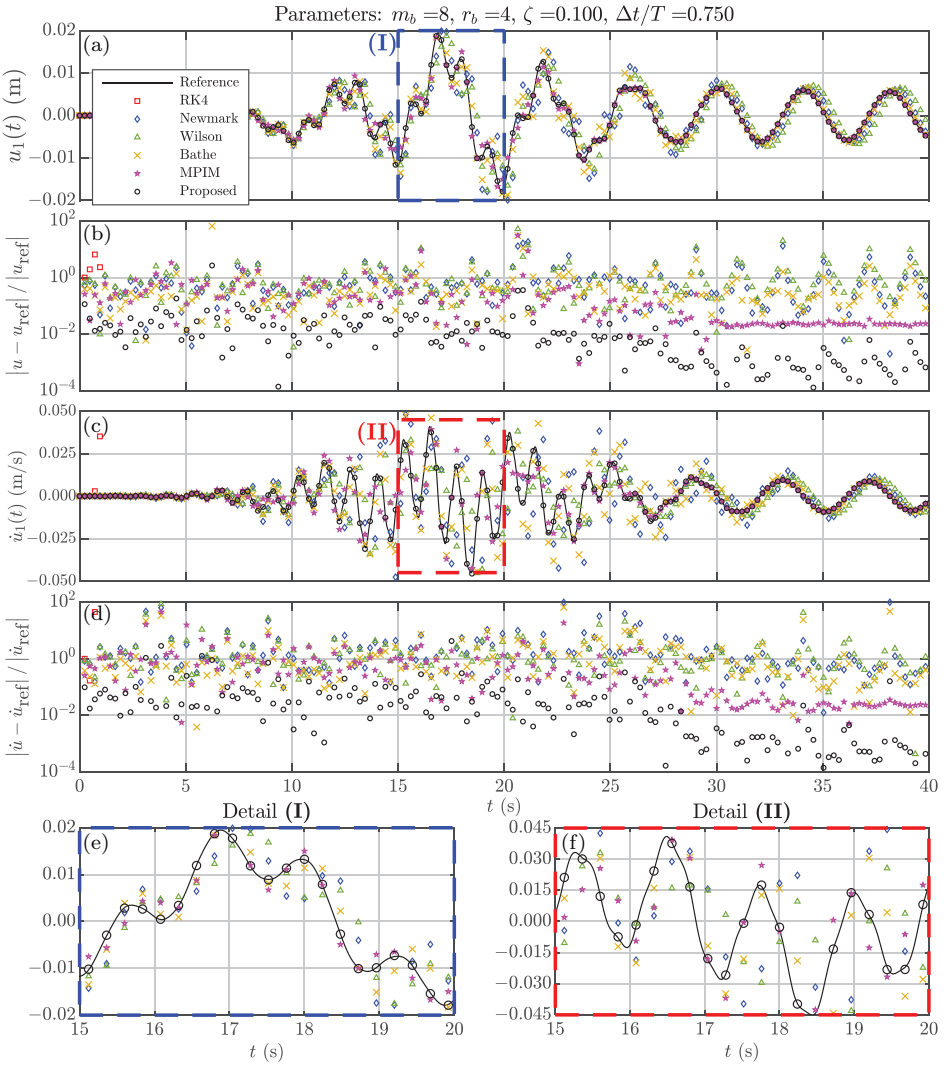

Fig. 3. Time-domain numerical solutions for the different methods at dof $u_1(t)$ with parmaeters $\zeta = 0.10$, time-step $\Delta t = 0.24$ s $= 0.75T$. Modified PIM (MPIM) has been implemented with $g = 4$ Gauss quadrature points for integration of nonhomogeneous term. Order of truncation in proposed method in the nonhomogeneous term: $m_b = 8$ and $r_b = 4$. (a) Displacement at dof-#1, $u_1(t)$, (b) local relative errors of displacements, (c) velocity at dof-#1, $\dot{u}_1(t)$, (d) local relative error of velocity, (e) and (f) detail plots of displacement and velocity in the range $15 \leq t \leq 20$ s.

In Fig. 3 the transient problem for the current example has been solved using the six different approaches presented above. Figures 3(a) and 3(c) show, respectively, the simulated response of the first degree of freedom, i.e., displacement $u_1(t)$ and the velocity $\dot{u}_1(t)$. The interval of simulation is $0 \leq t \leq t_{\max} = 40$ s, with a relative time-step of $\Delta t/T = 0.75$ s. Damping parameter has been taken $\zeta = 0.10$, something

that leads to a set of modal damping ratios in the range $0.97\% \leq \xi_j \leq 18.69\%$. The relationship between the spectral radius of matrices $\mathbf{M}^{-1}\mathbf{C}$ and $\sqrt{\mathbf{M}^{-1}\mathbf{K}}$, $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}}) = 0.815$ can be considered as a global measure of the dissipative forces. In order to calculate the relative errors of each numerical approach, a reference solution has been obtained using $\Delta t/500 = 4.8 \times 10^{-4}$ s. The relative errors along the time range have been plotted in Figs. 3(b) and 3(d). Since the chosen time step is relatively large, in fact $\Delta t/T \approx 0.75$, the methods based on the discretization of the derivatives (Newmark, Wilson and Bathe) are those ones that carry more numerical errors. The value of Δt used in Fig. 3 lies out of the stability region of the RK4 method. This can be observed at the first instants of the simulation, where the marker points of the RK4 (red squares) diverge. The MPIM and the proposed method have the best agreement with the reference solution. The parameter $m_b = 8$ has been used for this simulation, leading to a spectral radius $\rho(\boldsymbol{\beta}_b) = 0.8617 < 1$, guaranteeing the convergence although with a value very close to 1. We take $r_b = 4$ terms in the Neumann series noticing that from this value the errors do not become lower. The number of Gauss integration points for the MPIM are $g = 4$, revealing also that for $g > 4$ the accuracy barely changes. Under this choice of the parameters, better results are perceived with the proposed method, reaching relative errors between one and two orders of magnitude lower, both in displacements and velocities. In Figs. 3(e) and 3(f), detail windows of the solution within $15 \leq t \leq 20$ s are shown. In them, a very satisfactory agreement of the proposed approach is perceived compared with the rest of the simulated methods. The differences are especially evident in the evaluation of velocities, where for this relatively high time step, the other approaches show errors significantly higher than those recorded with the proposed method.

It is expected that a reduction of the time step will induce an improvement in the numerical solution for all methods. This effect is shown in Fig 4 where the displacement and velocity response together with the corresponding relative errors have been plotted for a 10-times smaller time step with respect to that used for Fig. 3 simulations, i.e., $\Delta t = 0.024 = 0.075T$. The damping level is left as the same value as before, keeping the value of the dissipative parameter $\zeta = 0.10$. With a time step 10 times smaller, and no change in damping, a reduction in the truncation error emerges as expected. Therefore, in order to computationally optimize the proposed algorithm, a smaller value of the truncation parameter m_b can be taken. Results of Fig. 4 have been simulated using the orders $m_b = 4$ and $r_b = 4$. In general, the relative errors drop by 2–4 orders of magnitude with respect to those shown in Fig. 3. Moreover, the RK4 method becomes now stable and shows very accurate results, even better than those of the MPIM. In the proposed method, the reduction of the time step leads to a substantial improvement of the predictions. Along the entire range of time simulation, the proposed approach is visibly the most accurate, among all the simulated approaches. The error in both displacement and velocity improve appreciably in all methodologies within the range $30 < t < 40$ s, where the nonhomogeneous terms are reduced to practically zero.

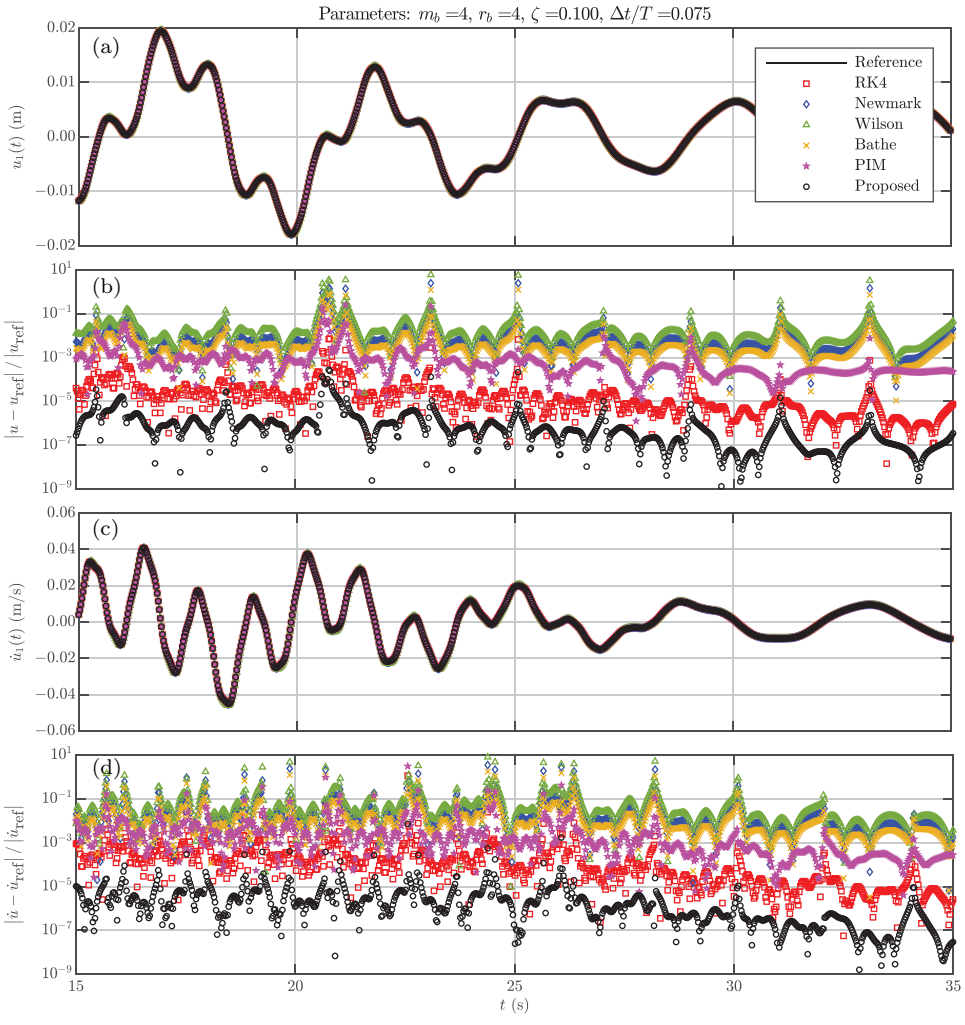

Fig. 4. Simulation of time-domain response of dof-1. Dimensionless damping parameter $\zeta = 0.10$. Time-step $\Delta t = 0.024 = 0.075T$ s. Modified PIM (MPIM) uses $g = 4$ Gauss quadrature points in integration of nonhomogeneous term. Order of truncation in proposed method in the nonhomogeneous term: $r_b = 4$, $m_b = 4$. (a) Displacement, $u_1(t)$, (b) relative error of displacement response, (c) velocity, $\dot{u}_1(t)$, (d) relative error of velocity response.

We will now focus on discussing the influence of different parameters in the global error. Accuracy of the proposed method mainly depends on the following three relevant parameters:

(i) The time step, Δt, measured as usual in dimensionless form respect to the minimum period of the system. For the current example $T = 2\pi/\omega_{\max} = 0.32$ s.

(ii) The damping level, quantified throughout the entries of the matrix $\mathbf{C}$. There are several forms of measuring how strong or light are the dissipative forces. In this example the coefficients of the dampers are proportional to the dimensionless parameter ζ. In the results the dimensionless ratio $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}})$ will be used as a relative dimensionless measure of the damping level.

(iii) The truncation orders, m_b and r_b, introduced to construct the nonhomogeneous term $\mathbf{b}_k(\Delta t)$ in Eqs. (61) and (63).

Let us define a measure of the total error (global error) over the entire time range. Given an interval $[0, t_{\max}]$, a time step Δt and a function $y(t)$ defined at all points $y(t_k)$, $0 \leq k \leq k_{\max}$, $t_k = k\Delta t$, with $k_{\max} = t_{\max}/\Delta t$. Then we define the following function-norm $\|\bullet\|$ as

$$\|y(t)\| = \sqrt{\sum_{k=0}^{k_{\max}} y^2(t_k)}, \tag{69}$$

and the global error of the jth dof displacement and velocity can then be defined as

$$e(u_j, \Delta t) = \frac{\|u_j(t) - u_{j,\mathrm{ref}}(t)\|}{\|u_{j,\mathrm{ref}}(t)\|}, \quad e(\dot{u}_j, \Delta t) = \frac{\|\dot{u}_j(t) - \dot{u}_{j,\mathrm{ref}}(t)\|}{\|\dot{u}_{j,\mathrm{ref}}(t)\|}. \tag{70}$$

The norm defined above will be employed as global measure of the error for each simulation and for each numerical method. The relationship between the global error and the time step is shown in Figs. 5(a) and 5(b) show, respectively, the errors $e(u_2, \Delta t)$ (displacement) and $e(\dot{u}_2, \Delta t)$ (velocity) at degree of freedom #2, for a relatively low damping, given by a $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}}) = 0.10$, corresponding to $\zeta = 0.0123$. The left-side plots (low damping) can be compared with those of the right side (high damping), Figs. 5(c) and 5(d), which show the same variables but for a higher level of damping, i.e., $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}}) = 1.00$ (obtained with $\zeta = 0.123$). In view of the results, the proposed method shows evidence of high accuracy compared to the other methods studied. Moreover, only the proposed method seems to be sensitive to the changes of the damping level. The plots shown for the rest of the approaches are hardly changed after increasing the damping level. The RK4 method shows high accuracy in the simulations, specially for low time step. However, this method exhibits a limitation by stability, something that can be observed in Fig. 5, where the RK4 becomes unstable approximately for $\Delta t/T > 0.2$. The loss of accuracy shown by the proposed method approximately at $\Delta t/T \approx 1$ is not due to instability but to divergence of the sum of the infinite series. As shown in Sec. 4, the spectral radius $\rho(\boldsymbol{\beta})$ is strongly dependent on the time-step and on the damping level. The MPIM has been simulated for different integration points $g = \{3, 4, 5\}$. However, it is appreciated that the global error is barely sensitive to this parameter. In fact, curves for $g = 3$, $g = 4$ and $g = 5$ show slight differences between each other only for high time steps.

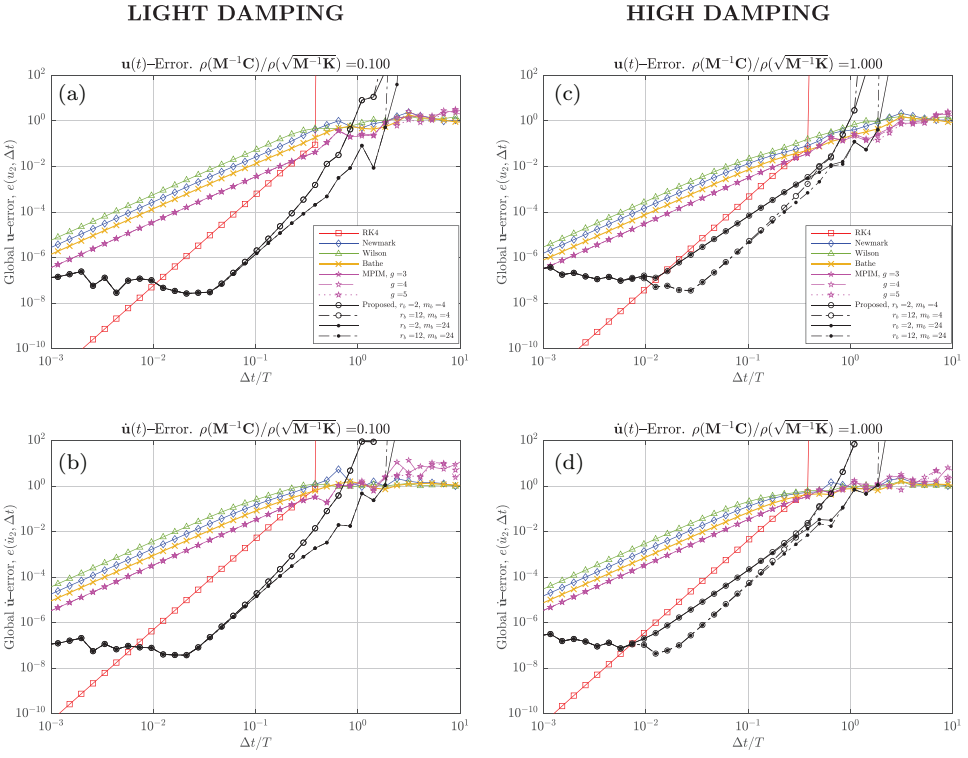

Fig. 5. Relationship between global error of dof # 2, $e(u_2, \Delta t)$,$e(\dot{u}_2, \Delta t)$ and time step $\Delta t/T$ (example 1), for two cases of damping level: (a) and (b) global error plots for $\zeta = 0.012$, $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}}) = 0.10$. (c) and (d) global error plots for $\zeta = 0.123$, $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}}) = 1.00$. Orders of truncation for the proposed method in the nonhomogeneous term: $m_b \in \{4, 24\}$ and $r_b \in \{2, 12\}$.

As expected, the error increases with the time step for all methods. However, for small values of $\Delta t/T$, the global error due to the proposed method is shown to be approximately insensitive to the time step. For the range of values $\Delta t/T$ used in practice, the proposed method is several orders of magnitude more accurate than the other schemes. Furthermore, as Δt increases, the error can be somehow reduced by increasing the parameters r_b and m_b (truncation orders in the computation of $\mathbf{b}(\Delta t)$). In order to identify the influence of these parameters, different simulations have been carried out for $r_b = \{2, 12\}$ and $m_b = \{4, 24\}$. An increase in these latter improves significantly the estimation, as can be seen comparing the results from $r_b = 2$ to $r_b = 12$ or those from $m_b = 4$ to $m_b = 24$. The results are specially sensitive as the damping increases, as noticed in Figs. 5(c) and 5(d).

The trend observed in Fig. 5 makes us wonder how the quality of our proposal will evolve as we increase the damping level. To this end, Fig. 6 shows the dependency of the global error with the damping intensity, measured by the magnitude $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}})$. Left and right side plots of Fig. 6 represent the relative

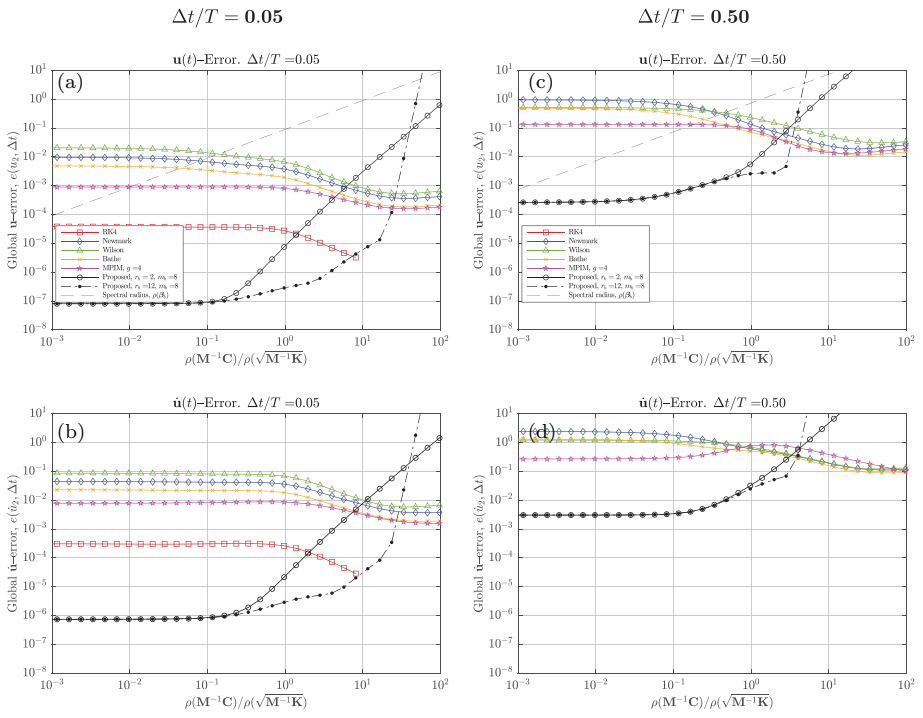

Fig. 6. Relationship between global error of dof # 2, $e(u_2, \Delta t)$,$e(\dot{u}_2, \Delta t)$ and damping level, measured as $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}})$ (example 1), for two time steps: (a) and (b) global error plots for $\Delta t/T = 0.05$. (c) and (d) global error plots for $\Delta t/T = 0.5$. Orders of truncation for the proposed method in the nonhomogeneous term: $m_b = 8$ and $r_b \in \{2, 12\}$. Dashed black line (no markers) in global error of displacements denotes the spectral radius of $\boldsymbol{\beta}_b$

global error for two time steps, $\Delta t/T = \{0.05, 0.50\}$, respectively. In all performed simulations, the proposed method still exhibits the most accurate results for lightly damped systems, say within the range $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}}) < 1$. In Figs 6(a) and 6(c) the spectral radius $\rho(\boldsymbol{\beta}_b)$ has also been plotted, showing a linear dependency with damping. Thus, in order to keep the errors bounded, the Neumann series needs to be updated with more terms as the damping increases. From the limit $\rho(\boldsymbol{\beta}_b) = 1$ it is useless to increase r_b since the Neumann series is divergent. It can be seen in the four graphs of fig. 6 the divergence of the method when this threshold is exceeded. The other methods under study are also insensitive to damping at low values of $\rho(\mathbf{M}^{-1}\mathbf{C})$. However, above certain limiting values, damping improves the accuracy of the result slightly, contrary to what is observed with the current approach. Despite this improvement, the damping affects the stability conditions for the RK4 method and the accuracy for the MPIM. For the time step $\Delta t/T = 0.05$, the RK4 method becomes unstable for $\rho(\mathbf{M}^{-1}\mathbf{C})/\rho(\sqrt{\mathbf{M}^{-1}\mathbf{K}}) \approx 8$. On the right plots, for $\Delta t/T = 0.5$, the RK4 is unstable in the whole range of damping level and MPIM exhibits a pronounced loss of accuracy for high damping levels.

Newmark, Wilson and Bathe methods show decreasing errors as the damping level is increased. Effects on the global error of multiplying the time step by 10 can be observed in Fig. 6. Indeed, from the left plots with $\Delta t/T = 0.05$ in Figs. 6(a) and 6(b) to the right plots with $\Delta t/T = 0.50$ in Figs. 6(c) and 6(d). As expected, all methods exhibit an increasing error, although the proposed approach for these particular values still yields the most precise numerical results for light and moderate damping problems.

In theory, the convergence of the method is conditioned to the time step and damping level. However, the cases covered in the simulations shown in Figs. 5 and 6 enable convergence to be guaranteed in most practical cases. Moreover, in the usual case of lightly damped structures, the parameter $\Delta t/T$ can be increased significantly without significant loss of accuracy. Furthermore, transient problems with very high damping can also be analyzed under the current approach by reducing the time step and increasing the truncation orders.

8. Conclusions

This paper considers the numerical resolution of transient problems in linear structural dynamics. Taking advantage that dissipative forces are in general much smaller than inertia and elastic forces, we propose to artificially perturb the damping terms to derive an iterative asymptotic procedure. The theoretical developments lead to the summation of the resulting infinite series culminating in an explicit iterative process. Convergence of the series, stability of the method and accuracy are studied in depth. Detailed algorithms for the computation of the main matrices are also presented. The proposed approach has been validated by means of a numerical example consisting of mass-spring-damper discrete multiple degree-of-freedom structure. The influence of the error with different parameters such as time step, damping level and numerical parameters intrinsic to the method is studied. The results are also compared with other existing methods in the literature, showing great accuracy both in displacements and velocities for a wide range of time steps.

Acknowledgments

This research was partially supported by the Grant PID2020-112759GB-I00 funded by MCIN/AEI/10.13039/501100011033. The author is also grateful for the partial support and funding of the open access from the grant CIGE/2021/141 funded by Generalitat Valenciana (Emerging Research Groups Grants) and by the European Social Fund.

Appendix A. Proof of Convergence Conditions

In this Appendix, the conditions for convergence will be discussed. As derived in Sec. 3, the transient solution for displacements and velocities at instant $t = t_k$ is

the result of the series

$$\mathbf{U}_k = \begin{Bmatrix} \mathbf{u}_k \\ \dot{\mathbf{u}}_k \end{Bmatrix} \equiv \lim_{n\to\infty} \mathbf{U}_k^{(n)} = \sum_{n=0}^{\infty} \begin{Bmatrix} \mathbf{x}_k^{(n)} \\ \dot{\mathbf{x}}_k^{(n)} \end{Bmatrix} = \sum_{n=0}^{\infty} \mathbf{X}_k^{(n)}, \quad k \geq 0. \tag{A.1}$$

The convergence of this series must be ensured for each $k \geq 1$. For this purpose, we will deduce a recursive relation in the space of the asymptotic expansion (n-space) that allows us to explicitly derive the sequence of partial sums. The sequence $\{\mathbf{X_k}^{(n)}\}_{n=0}^{\infty}$ can be determined from the double recursive relation given by Eq. (28), which is rewritten here as

$$\mathbf{X}_{k+1}^{(n)} = \mathbf{T}\,\mathbf{X}_k^{(n)} + \boldsymbol{\alpha}\,\mathbf{X}_k^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_{k+1}^{(n-1)}, \quad n \geq 1, \quad k \geq 0, \tag{A.2}$$

Evaluating this equation for $n \geq 2$ and for $k = \{0, 1, \ldots, k-1\}$ yields

$$\begin{aligned} \mathbf{X}_1^{(n)} &= \mathbf{T}\,\mathbf{X}_0^{(n)} + \boldsymbol{\alpha}\,\mathbf{X}_0^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_1^{(n-1)}, \\ \mathbf{X}_2^{(n)} &= \mathbf{T}\,\mathbf{X}_1^{(n)} + \boldsymbol{\alpha}\,\mathbf{X}_1^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_2^{(n-1)}, \\ &\vdots = \vdots \\ \mathbf{X}_k^{(n)} &= \mathbf{T}\,\mathbf{X}_{k-1}^{(n)} + \boldsymbol{\alpha}\,\mathbf{X}_{k-1}^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_k^{(n-1)}, \quad n \geq 2, \end{aligned} \tag{A.3}$$

Since $\mathbf{X}_0^{(n)} = \mathbf{X}_0^{(n-1)} = \mathbf{0}$ for $n \geq 2$, then the first step ($k = 0$) is reduced to $\mathbf{X}_1^{(n)} = \boldsymbol{\beta}\,\mathbf{X}_1^{(n-1)}$. The terms $\mathbf{X}_1^{(n)}, \ldots, \mathbf{X}_{k-1}^{(n)}$ of the right side of each step equation can be replaced by their corresponding expression of the previous step. Repeating this process systematically for each step and after some operations, Eq. (A.3) can be written separating the terms associated to the nth and $(n-1)$th iterations in both sides of equations, i.e.,

$$\begin{aligned} \mathbf{X}_1^{(n)} &= \boldsymbol{\beta}\,\mathbf{X}_1^{(n-1)}, \\ \mathbf{X}_2^{(n)} &= \boldsymbol{\gamma}\mathbf{X}_1^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_2^{(n-1)}, \\ \mathbf{X}_3^{(n)} &= \mathbf{T}\boldsymbol{\gamma}\mathbf{X}_1^{(n-1)} + \boldsymbol{\gamma}\,\mathbf{X}_2^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_3^{(n-1)}, \\ &\vdots = \vdots \\ \mathbf{X}_k^{(n)} &= \sum_{r=1}^{k-1} \mathbf{T}^{k-1-r}\,\boldsymbol{\gamma}\,\mathbf{X}_r^{(n-1)} + \boldsymbol{\beta}\,\mathbf{X}_k^{(n-1)}, \quad n \geq 2, \end{aligned} \tag{A.4}$$

where $\boldsymbol{\gamma} = \mathbf{T}\boldsymbol{\beta} + \boldsymbol{\alpha}$. Rearranging $\mathbf{X}_1^{(n)}, \ldots, \mathbf{X}_k^{(n)}$ in one single $2Nk$–size column vector denoted by $\mathbf{X}^{(n)} \in \mathbb{R}^{2Nk}$, and after some algebra, the above relations can be expressed as

$$\mathbf{X}^{(n)} = \mathbf{R}\,\mathbf{X}^{(n-1)}, \quad n \geq 2, \tag{A.5}$$

where the nth iteration array $\mathbf{X}^{(n)}$ and matrix $\mathbf{R}$ are formed by the blocks

$$\mathbf{X}^{(n)} = \begin{Bmatrix} \mathbf{X}_1^{(n)} \\ \vdots \\ \mathbf{X}_k^{(n)} \end{Bmatrix} \in \mathbb{R}^{2Nk},$$

$$\mathbf{R} = \begin{bmatrix} \boldsymbol{\beta} & \mathbf{0}_{2N} & \mathbf{0}_{2N} & \cdots & \mathbf{0}_{2N} & \mathbf{0}_{2N} \\ \boldsymbol{\gamma} & \boldsymbol{\beta} & \mathbf{0}_{2N} & \cdots & \mathbf{0}_{2N} & \mathbf{0}_{2N} \\ \mathbf{T}\boldsymbol{\gamma} & \boldsymbol{\gamma} & \boldsymbol{\beta} & \cdots & \mathbf{0}_{2N} & \mathbf{0}_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{T}^{k-3}\boldsymbol{\gamma} & \mathbf{T}^{k-4}\boldsymbol{\gamma} & \mathbf{T}^{k-5}\boldsymbol{\gamma} & \cdots & \boldsymbol{\beta} & \mathbf{0}_{2N} \\ \mathbf{T}^{k-2}\boldsymbol{\gamma} & \mathbf{T}^{k-3}\boldsymbol{\gamma} & \mathbf{T}^{k-4}\boldsymbol{\gamma} & \cdots & \boldsymbol{\gamma} & \boldsymbol{\beta} \end{bmatrix} \in \mathbb{R}^{2Nk\times 2Nk}. \tag{A.6}$$

The first iteration for $n = 1$ leads to the relationship between $\mathbf{X}^{(0)}$ and $\mathbf{X}^{(1)}$. It can be derived straightforward when Eq. (A.3) is evaluated at $n = 1$ and taking into account that $\mathbf{X}_0^{(1)} = \mathbf{0}$ and $\mathbf{X}_0^{(0)} = \mathbf{U}_0$. Indeed,

$$\begin{aligned} \mathbf{X}_1^{(1)} &= \mathbf{T}\,\mathbf{X}_0^{(1)} + \boldsymbol{\alpha}\,\mathbf{X}_0^{(0)} + \boldsymbol{\beta}\,\mathbf{X}_1^{(0)} = \boldsymbol{\beta}\,\mathbf{X}_1^{(0)} + \boldsymbol{\alpha}\,\mathbf{U}_0, \\ \mathbf{X}_2^{(1)} &= \mathbf{T}\,\mathbf{X}_1^{(1)} + \boldsymbol{\alpha}\,\mathbf{X}_1^{(0)} + \boldsymbol{\beta}\,\mathbf{X}_2^{(0)} = \mathbf{T}\,\boldsymbol{\gamma}\,\mathbf{X}_1^{(0)} + \boldsymbol{\beta}\,\mathbf{X}_2^{(0)} + \mathbf{T}\,\boldsymbol{\alpha}\,\mathbf{U}_0, \\ &\vdots \quad \vdots \\ \mathbf{X}_k^{(1)} &= \mathbf{T}\,\mathbf{X}_{k-1}^{(1)} + \boldsymbol{\alpha}\,\mathbf{X}_{k-1}^{(0)} + \boldsymbol{\beta}\,\mathbf{X}_k^{(0)} \\ &= \sum_{r=1}^{k-1} \mathbf{T}^{k-1-r}\,\boldsymbol{\gamma}\,\mathbf{X}_r^{(0)} + \boldsymbol{\beta}\,\mathbf{X}_k^{(0)} + \mathbf{T}^{k-1}\,\boldsymbol{\alpha}\,\mathbf{U}_0, \quad n \geq 2. \end{aligned} \tag{A.7}$$

As before, using some matrix algebra, the following expression can be obtained:

$$\mathbf{X}^{(1)} = \mathbf{R}\,\mathbf{X}^{(0)} + \mathbf{S}\,\mathbf{U}_0, \tag{A.8}$$

where

$$\mathbf{S} = \begin{bmatrix} \boldsymbol{\alpha} \\ \mathbf{T}\,\boldsymbol{\alpha} \\ \vdots \\ \mathbf{T}^{k-1}\,\boldsymbol{\alpha} \end{bmatrix} \in \mathbb{R}^{2Nk\times 2N}. \tag{A.9}$$

We can build the $2Nk$–size column vector $\mathbf{U}^{(n)}$, result of the partial sums of the first n steps of the asymptotic scheme

$$\mathbf{U}^{(n)} = \begin{Bmatrix} \mathbf{U}_1^{(n)} \\ \vdots \\ \mathbf{U}_k^{(n)} \end{Bmatrix} = \begin{Bmatrix} \sum_{\nu=0}^{n} \mathbf{X}_1^{(\nu)} \\ \vdots \\ \sum_{\nu=0}^{n} \mathbf{X}_k^{(n)} \end{Bmatrix} = \sum_{\nu=0}^{n} \begin{Bmatrix} \mathbf{X}_1^{(\nu)} \\ \vdots \\ \mathbf{X}_k^{(\nu)} \end{Bmatrix} = \sum_{\nu=0}^{n} \mathbf{X}^{(\nu)}. \tag{A.10}$$

Using the recursive relations of Eq. (A.5) for $n \geq 2$ together with the first iteration ($n = 1$) from Eq. (A.8), the response $\mathbf{U}^{(n)}$ results

$$\begin{aligned} \mathbf{U}^{(n)} &= \mathbf{X}^{(0)} + \mathbf{X}^{(1)} + \mathbf{X}^{(2)} + \cdots + \mathbf{X}^{(n)} \\ &= \mathbf{X}^{(0)} + \mathbf{X}^{(1)} + \mathbf{R}\mathbf{X}^{(1)} + \cdots + +\mathbf{R}^{n-1}\mathbf{X}^{(1)} \\ &= \mathbf{X}^{(0)} + \left(\mathbf{I}_{2Nk} + \mathbf{R} + \cdots + +\mathbf{R}^{n-1}\right)\mathbf{X}^{(1)} \\ &= \mathbf{X}^{(0)} + \left(\mathbf{I}_{2Nk} + \mathbf{R} + \cdots + +\mathbf{R}^{n-1}\right)\left(\mathbf{R}\,\mathbf{X}^{(0)} + \mathbf{S}\,\mathbf{U}_0\right) \\ &= \left(\sum_{\nu=0}^{n} \mathbf{R}^{\nu}\right)\mathbf{X}^{(0)} + \left(\sum_{\nu=0}^{n-1} \mathbf{R}^{\nu}\right)\mathbf{S}\,\mathbf{U}_0. \end{aligned} \tag{A.11}$$

The asymptotic solution considering the infinite perturbation steps, i.e., $n \to \infty$ is then

$$\begin{aligned} \mathbf{U} &= \lim_{n\to\infty} \mathbf{U}^{(n)} = \lim_{n\to\infty} \sum_{\nu=0}^{n} \mathbf{X}^{(\nu)} \\ &= \lim_{n\to\infty} \left(\sum_{\nu=0}^{n} \mathbf{R}^{\nu}\right)\mathbf{X}^{(0)} + \lim_{n\to\infty} \left(\sum_{\nu=0}^{n-1} \mathbf{R}^{\nu}\right)\mathbf{S}\,\mathbf{U}_0. \end{aligned} \tag{A.12}$$

It is straightforward that the sequence of partial sums is

$$\left(\sum_{\nu=0}^{n-1} \mathbf{R}^{\nu}\right) = \left(\mathbf{I}_{2Nk} - \mathbf{R}\right)^{-1}\left(\mathbf{I}_{2Nk} - \mathbf{R}^{n}\right), \tag{A.13}$$

which is convergent provided that

$$\lim_{n\to\infty} \mathbf{R}^{n} = \mathbf{0}.$$

A proof of this result can be found in the book of (Wilkinson [1988], Sec. 55, p. 59) and in the book of (Householder [1964], Sec. 2.5, p. 54). Therefore, the necessary and sufficient condition for this series to be convergent is that all the eigenvalues of matrix $\mathbf{R}$ are strictly smaller than the unit in absolute value [Householder (1964); Wilkinson (1988)]. Therefore, the series is convergent provided that its spectral radius is less than the unity, i.e., $\rho(\mathbf{R}) < 1$. The matrix $\mathbf{R}$ is a triangular matrix by blocks as shown in Eq. (A.6). The main block diagonal is formed by the repetition

of matrix $\boldsymbol{\beta}$. Consequently, the eigenvalues of $\mathbf{R}$ are those of $\boldsymbol{\beta}$ and therefore $\rho(\mathbf{R}) = \rho(\boldsymbol{\beta})$. Hence, the convergence of the method can be ensured provided that

$$\rho(\boldsymbol{\beta}) < 1. \tag{A.14}$$

Under this condition the sequence of power series $\{\mathbf{R}^{\nu}\}_{\nu=0}^{\infty}$ tends to zero and a closed form expression of the solution can be written as

$$\mathbf{U} = (\mathbf{I}_{2Nk} - \mathbf{R})^{-1} \left(\mathbf{X}^{(0)} + \mathbf{S}\,\mathbf{U}_0\right). \tag{A.15}$$

This closed-form relationship highlights the intrinsic nature of the proposed approach as an asymptotic expansion-based method. Note that both matrices $\mathbf{R}$ and $\mathbf{S}$ are directly proportional to the damping matrix, so that for undamped problems we have $\mathbf{U} \equiv \mathbf{X}^{(0)}$ (undamped solution).

References

Bathe, K.-J. [2007] "Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration scheme," *Comput. Struct.* **85**(7), 437–445, doi:10.1016/j.compstruc.2006.09.004.

Bathe, K.-J. [2014] *Finite Element Procedures in Engineering Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1982.

Bathe, K.-J. and Baig, M. [2005] "On a composite implicit time integration procedure for nonlinear dynamics," *Comput. Struct.* **83**(31–32), 2513–2524, doi:10.1016/j.compstruc.2005.08.001.

Bathe, K.-J. and Noh, G. [2012] "Insight into an implicit time integration scheme for structural dynamics," *Comput. Struct.* **98–99**, 1–6, doi:10.1016/j.compstruc.2012.01.009.

Berrahma-Chekroun, N., Fafard, M. and Gervais, J. J. [2001] "Resolution of the transient dynamic problem with arbitrary loading using the asymptotic method," *J. Sound Vib.* **243**(3), 475–501, doi:10.1006/jsvi.2000.3423.

Burden, R. L. and Faires, J. D. [2001] *Numerical Analysis* (Brooks-Cole, Cengage Learning), doi:10.4324/9781315154039-12.

Butcher, J. [2008] *Numerical Methods for Ordinary Differential Equations*, 2nd Edition (John Wiley and Sons), doi:10.1002/9780470753767.

Caprani, C. C. [2013] "A modal precise integration method for the calculation of footbridge vibration response," *Comput. Struct.* **128**, 116–127, doi:10.1016/j.compstruc.2013.06.006.

Cha, P. D. [2005] "Approximate eigensolutions for arbitrarily damped nearly proportional systems," *J. Sound Vib.* **288**(4–5), 813–827, doi:10.1016/j.jsv.2005.01.010.

Chung, K. and Lee, C. [1986] "Dynamic reanalysis of weakly non-proportionally damped systems," *J. Sound Vib.* **111**(1), 37–50, doi:10.1016/s0022-460x(86)81421-3.

Cochelin, B., Damil, N. and Potier-Ferry, M. [1994] "Asymptotic-numerical methods and pade approximants for nonlinear elastic structures," *Int. J. Numer. Methods Eng.* **37**(7), 1187–1213, doi:10.1002/nme.1620370706.

Daya, E. M. and Potier-Ferry, M. [2001] "A numerical method for nonlinear eigenvalue problems application to vibrations of viscoelastic structures," *Comput. Struct.* **79**(5), 533–541, doi:10.1016/s0045-7949(00)00151-6.

Duigou, L., Daya, E. M. and Potier-Ferry, M. [2003] "Iterative algorithms for non-linear eigenvalue problems. application to vibrations of viscoelastic shells," *Comput. Methods Appl. Mech. Eng.* **192**(11–12), 1323–1335, doi:10.1016/s0045-7825(02)00641-2.

Fafard, M., Henchi, K. and Gendron, G. [1997] "Application of an asymptotic method to transient dynamic problems," *J. Sound Vib.* **208**(1), 73–99, doi:10.1006/jsvi.1997.1169.

Fung, T. C. [1997] "A precise time-step integration method by step-response and impulsive-response matricesfor dynamic problems," *Int. J. Numer. Methods Eng.* **40**, 4501–4527, doi:10.1002/(SICI)1097-0207(19971230)40:24<4501::AID-NME266>3.0.CO;2-U.

Fung, T. C. [2004] "Computation of the matrix exponential and its derivatives by scaling and squaring," *Int. J. Numer. Methods Eng.* **59**(10), 1273–1286, doi:10.1002/nme.909.

Fung, T. C. and Chen, Z. L. [2006] "Krylov precise time-step integration method," *Int. J. Numer. Methods Eng.* **68**(11), 1115–1136, doi:10.1002/nme.1737.

Gallagher, R. H. [1975] *Perturbation Procedures in Nonlinear Finite Element Structural Analysis* (Springer, Berlin), pp. 75–89, doi:10.1007/bfb0074150.

Householder, A. S. [1964] *The Theory of Matrices in Numerical Analysis* (Dover Publications).

Lázaro, M. [2015] "Nonviscous modes of nonproportionally damped viscoelastic systems," *J. Appl. Mech.* (*Trans. ASME*) **82**(12), 121011, doi:10.1115/1.4031569.

Lázaro, M. [2016] "Eigensolutions of non-proportionally damped systems based on continuous damping sensitivity," *J. Sound Vib.* **363**(C), 532–544, doi:10.1016/j.jsv.2015.10.014.

Lázaro, M. [2018] "Closed-form eigensolutions of nonviscously, nonproportionally damped systems based on continuous damping sensitivity," *J. Sound Vib.* **413**, 368–382, doi:10.1016/j.jsv.2017.10.011.

Lázaro, M., Casanova, C. F., Ferrer, I. and Martín, P. [2016] "Analysis of nonviscous oscillators based on the damping model perturbation," *Shock Vib.* **2016**, 368129, doi:10.1155/2016/9634103.

Lázaro, M. and Pérez-Aparicio, J. L. [2013] "Dynamic analysis of frame structures with free viscoelastic layers: New closed-form solutions of eigenvalues and a viscous approach," *Eng. Struct.* **54**, 69–81, doi:10.1016/j.engstruct.2013.03.052.

Lin, J., Shen, W. and Williams, F. [1995] "A high precision direct integration scheme for structures subjected to transient dynamic loading," *Comput. Struct.* **56**(1), 113–120, doi:10.1016/0045-7949(94)00537-d.

Mansur, W., Loureiro, F., Soares, D. and Dors, C. [2007] "Explicit time-domain approaches based on numerical green functions computed by finite differences — the exga family," *J. Comput. Phys.* **227**(1), 851–870, doi:https://doi.org/10.1016/j.jcp.2007.08.024.

Mei, S.-L., Du, C.-J. and Zhang, S.-W. [2008] "Asymptotic numerical method for multi-degree-of-freedom nonlinear dynamic systems," *Chaos Solitons Fractals* **35**(3), 536–542, doi:10.1016/j.chaos.2006.05.067.

Meirovitch, L. and Ryland, G. [1985] "A perturbation technique for gyroscopic systems with small internal and external damping," *J. Sound Vib.* **100**(3), 393–408, doi:10.1016/0022-460x(85)90295-0, cited By 21.

Meirovitch, L. and Ryland, G, II. [1979] "Response of slightly damped gyroscopic systems," *J. Sound Vib.* **67**(1), 1–19, doi:10.1016/0022-460x(79)90497-8, cited By 39.

Moler, C. and Loan, C. V. [2003] "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Rev.* **45**(1), 3–49, doi:10.1137/s00361445024180.

Nayfeh, A. H. [2004] *Perturbation Methods* (Wiley-VCH), doi:10.1002/9783527617609.

Newmark, N. M. [1959] "A method of computation for structural dynamics," *J. Eng. Mech. Div.* **85**(EM3), 67–94, doi:10.1061/jmcea3.0000098.

Peres-Da-Silva, S., Cronin, D. and Randolph, T. [1995] "Computation of eigenvalues and eigenvectors of nonclassically damped systems," *Comput. Struct.* **57**(5), 883–891, doi:10.1016/0045-7949(95)00079-v.

Wan-Xie, Z. [2004] "On precise integration method," *J. Comput. Appl. Math.* **163**(1), 59–78, doi:10.1016/j.cam.2003.08.053.

Wang, M. and Au, F. T. K. [2007] "Precise integration method without inverse matrix calculation for structural dynamic equations," *Earthq. Eng. Eng. Vib.* **6**, 57–64, doi:10.1007/s11803-007-0661-2.

Wang, M. and Zhou, X. [2005] "Modified precise time step integration method of structural dynamic analysis," *Earthq. Eng. Eng. Vib.* **4**, 287–293, doi:10.1007/s11803-005-0011-1.

Wang, M. F. and Au, F. T. K. [2008] "Precise integration methods based on the Chebyshev polynomial of the first kind," *Earthq. Eng. Eng. Vib.* **7**, 207–216, doi:10.1007/s11803-008-0837-4.

Wang, M.-F. and Au, F. T. K. [2009] "Precise integration methods based on lagrange piecewise interpolation polynomials," *Int. J. Numer. Methods Eng.* **77**(7), 998–1014, doi:10.1002/nme.2444.

Wilkinson, J. H. [1988] *The Algebraic Eigenvalue Problem* (Oxford University Press).

Wilson, E., Farhoomand, I. and Bathe, K. [1973] "Nonlinear dynamic analysis of complex structures." *Earthq. Eng. Struct. Dyn.* **1**, 241–252, doi:10.1002/eqe.4290010305.

Zhong, W. X. and Williams, F. W. [1994] "A precise time step integration method," *Proc. Inst. Mech. Eng. C: J. Mech. Eng. Sci.* **208**(6), 427–430, doi:10.1243/pime_proc_1994_208_148_02.

Zhu, X. Q. and Law, S. S. [2001] "Precise time-step integration for the dynamic response of a continuous beam under moving loads," *J. Sound Vib.* **240**, 962–970, doi:10.1006/jsvi.2000.3184.

Zoghaib, L. and Mattei, P.-O. [2014] "Time and frequency response of structures with frequency dependent, non-proportional linear damping," *J. Sound Vib.* **333c**(3), 887–900, doi:10.1016/j.jsv.2013.09.044.