Damping perturbation based time integration asymptotic method for structural dynamics

Mario Lázaro\textsuperscript{a,b,*}

\textsuperscript{a}Department of Continuum Mechanics and Theory of Structures
Universitat Politècnica de València 46022 Valencia, Spain
\textsuperscript{b}Instituto Universitario de Matemática Pura y Aplicada
Universitat Politècnica de València 46022 Valencia, Spain

Abstract

The light damping hypothesis is usually assumed in structural dynamics since dissipative forces are in general weak with respect to inertial and elastic forces. In this paper a novel numerical method of time integration based on the artificial perturbation of damping is proposed. The asymptotic expansion of the transient response results in an infinite series which can be summed, leading to a well-defined explicit iterative step-by-step scheme. Conditions for convergence are rigorously analyzed, enabling the determination of the methodology boundaries in form of maximum time step. The numerical properties of the iterative scheme, i.e. stability, accuracy and computational effort are also studied in detail. The approach is validated with two numerical examples, showing a high accuracy and computational efficiency relative to other methods.

Keywords: time integration, perturbation method, viscous damping, structural dynamics, transient problem, iterative algorithm, asymptotic expansion

1. Introduction

In this paper, the 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

\[
M \ddot{u} + C \dot{u} + K u = f(t), \quad u(0) = u_0, \quad \dot{u}(0) = \dot{u}_0
\]

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

Time-integrator algorithms have become an essential tool in the numerical simulation of structural dynamics. In general, two families of methods can be distinguished: explicit and implicit schemes \[1, 2\]. In the former, the response at each time step is evaluated from the previous step. Implicit integration requires the solution of one or more systems of linear equations at each iteration. In linear problems, the coefficient matrices are time invariant and a proper factorization of these allows to optimize the solution of the response at each step. Most implicit algorithms exhibit unconditional stability: Newmark’s method \[3\], Wilson’s method \[4\], Hilber-Hughes-Taylor’s method (HHT) \[5\], Wood-Bozak-ZienKiewicz’s method (WBZ) \[6\]. The HHT and WBZ approaches were proposed with the aim of introducing an artificial dissipation in
order to eliminate the high-frequency part of the response. As consequence, damping inconsistencies arose, which were addressed by the \( \alpha \)-scheme from Chung and Hulbert \[7\]. Krenk \[8\] presented a full energy analysis of Newmark based time integration algorithms. The Bathe’s method \[9–11\] proposes a composite time-integration approach, splitting up the time-step into two sub-steps, so that the trapezoidal rule is used for the first sub-step and the 3-point Euler backward method for the second one. The Bathe’s method has been recently improved with the introduction of new parameters as a controllable spectral radius \[12–13\] or a prescribing desired numerical dissipation \[14\].

The computational effort of implicit methods increases as the number of degrees of freedom of the system becomes larger. In such cases, explicit methods are in general more competitive. However, a known drawback of explicit methods is their conditional stability, such as the central difference method (CDM) or the 4th order Runga-Kutta method (RK4)\[15, 16\]. 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 \[17, 18\], 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 \[19\]. 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 non-homogeneous terms in the equations. Zhong and Williams \[17\] 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 \[20\] improved the PIM by developing in trigonometric Fourier series the external force term within the time step. Wang et al. \[21\] proposed the so-called Modified Precise Integration Method (MPIM), which consisted of evaluating the non-homogeneous 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 \[22\] presented a PIM avoiding computation of the inverse matrix for the evaluation of the non-homogeneous terms. They proposed two methods for the evaluation of the independent term based on interpolation with Chebyshev \[23\] and Lagrange polynomials \[24\]. Fung \[25\] 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 \[26\] 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 \[27\], the PIM is applied for a continuous Euler-Bernouilli beam under moving loads. Caprani \[28\] 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 priori to be very small respect to other terms of the model \[29\]. 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 non-classically viscously \[30–33\] and non-viscously \[34–42\] damped structures. In the time domain, techniques based on asymptotic perturbation are very useful to obtain solutions in nonlinear mechanics \[43–45\]. In linear structural dynamics, Fafard et al. \[46\] and Berhama-Chekroun \[47\] 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). Perhaps 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, thus avoiding double iteration. Conditions for
convergence of the method have been explored allowing the determination boundary limits of allowable dynamic range and time-step. Algorithms for the computation of the main matrices of the scheme are developed. Stability, accuracy and computational effort are addressed in detail. The proposed method is validated through two numerical examples, covering discrete and continuous structures. 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 methods. Moreover, the computational time is very competitive with respect to other explicit methods, such as Modified PIM.

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, the terms $C\dot{u}$ of Eqs. (1) can be considered as a perturbation of the undamped case, something that can be reproduced by an artificial parameter. We propose to modify Eq. (1) with the parameter $\epsilon$, with $0 \leq \epsilon \leq 1$, which multiplies the damping matrix in Eq. (1) resulting in:

$$M\ddot{u} + \epsilon C\dot{u} + Ku = f(t)$$

(2)

Now, the solution of Eq. (2) is a two-variable vector $u(t, \epsilon)$. The $\epsilon$-dependent solution can be expanded as

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

(3)

where $\{x^{(n)}(t)\}_{n=0}^{\infty} \in \mathbb{R}^N$ 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

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

(4)

At the lower limit of the range, $\epsilon = 0$, it is clear that we are in front of the undamped problem

$$u(t, 0) = x^{(0)}(t)$$

(5)

The initial conditions are satisfied by the first iteration $x^{(0)}(t)$, so that $x^{(0)}(0) = u_0$ and $\dot{x}^{(0)}(0) = \dot{u}_0$. Thus, the problem for $n = 0$ can be written as

$$\begin{align*}
M\ddot{x}^{(0)} + Kx^{(0)} &= f(t) \\
x^{(0)}(0) &= u_0, \quad \dot{x}^{(0)}(0) = \dot{u}_0
\end{align*}$$

(6)

Substituting Eq. (4) into Eq. (2) and identifying the coefficients of $\epsilon^n$, the $n$th function can be determined. After some straight operations it yields

$$\begin{align*}
M\ddot{x}^{(n)} + Kx^{(n)} &= -C\dot{x}^{(n-1)} \\
x^{(n)}(0) &= 0, \quad \dot{x}^{(n)}(0) = 0, \quad n \geq 1
\end{align*}$$

(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. (4). 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 \( \Delta t \), 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 \( X_k^{(0)} = X^{(0)}(t_k) \). A recursive scheme is built using the explicit method based on Green’s functions \([48]\) yielding

\[
X_{k+1}^{(0)} = G(\Delta t)X_k^{(0)} + H(\Delta t)\dot{X}_k^{(0)} + \int_{t=t_k}^{t_{k+1}} H(t_{k+1} - t)M^{-1}f(t)\,dt
\]

and

\[
\dot{X}_{k+1}^{(0)} = \dot{G}(\Delta t)X_k^{(0)} + \dot{H}(\Delta t)\dot{X}_k^{(0)} + \int_{t=t_k}^{t_{k+1}} G(t_{k+1} - t)M^{-1}f(t)\,dt
\]

The Green’s functions, \( G(t) \) and \( H(t) \), are the solutions of the following system of matrix differential equations

\[
\begin{cases}
\dot{G}(t) = -A \, H(t), & H(0) = 0 \\
\dot{H}(t) = G(t), & G(0) = I_N
\end{cases}
\]

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

\[
G(t) = \cos(\sqrt{A} t) = \sum_{j=0}^{\infty} \frac{(-1)^j A^j}{(2j)!} t^{2j}, \quad H(t) = A^{-1/2} \sin(\sqrt{A} t) = \sum_{j=0}^{\infty} \frac{(-1)^j A^j}{(2j + 1)!} t^{2j+1}
\]

Sometimes \( G(t) \) and \( H(t) \) are named step-response and impulsive-response functions \([25]\), 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

\[
f(t) \approx \sum_{i=1}^{4} L_i \left( \frac{t - t_k}{\Delta t} \right) f \left( t_k + \frac{(i - 1)\Delta t}{3} \right), \quad t_k \leq t \leq t_{k+1}
\]

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

\[
L_1(\xi) = \frac{1}{2}(1 - \xi)(1 - 2\xi)(1 - 3\xi), \quad L_2(\xi) = \frac{9}{2}(1 - \xi)(1 - 2\xi)\xi, \quad L_3(\xi) = -\frac{9}{2}(1 - \xi)(1 - 3\xi)\xi, \quad L_4(\xi) = \frac{1}{2}(1 - 3\xi)(2 - 3\xi)\xi
\]

Plugging Eq. (10) into the integrals of Eq. (6) and after some operations we obtain

\[
\int_{t=t_k}^{t_{k+1}} H(t_{k+1} - t)M^{-1}f(t)\,dt \approx \sum_{i=1}^{4} L_{ui} M^{-1} f \left( t_k + \frac{(i - 1)\Delta t}{3} \right)
\]

and

\[
\int_{t=t_k}^{t_{k+1}} G(t_{k+1} - t)M^{-1}f(t)\,dt \approx \sum_{i=1}^{4} L_{vi} M^{-1} f \left( t_k + \frac{(i - 1)\Delta t}{3} \right)
\]

where

\[
L_{ui} = \int_{t=t_k}^{t_{k+1}} H(t_{k+1} - t) L_i \left( \frac{t - t_k}{\Delta t} \right) dt, \quad L_{vi} = \int_{t=t_k}^{t_{k+1}} G(t_{k+1} - t) L_i \left( \frac{t - t_k}{\Delta t} \right) dt, \quad 1 \leq i \leq 4
\]

Rearranging together displacements and velocities of the degrees of freedom in the same \( 2N \)-size vector, the iterative process can be summarized as the following relationship

\[
X_{k+1}^{(0)} = T X_k^{(0)} + L g_k, \quad k \geq 0
\]
where

\[ X_k^{(0)} = \left\{ x_k^{(0)}, x_k^{(1)} \right\} \in \mathbb{R}^{2N}, \quad X_0^{(0)} = \left\{ u_0, u_0 \right\}, \quad T = \begin{bmatrix} G(\Delta t) & H(\Delta t) \\ G(\Delta t) & H(\Delta t) \end{bmatrix} \in \mathbb{R}^{2N \times 2N} \]

and

\[ L = \begin{bmatrix} L_{u1} & L_{u2} & L_{u3} & L_{u4} \\ L_{u1} & L_{u2} & L_{u3} & L_{u4} \end{bmatrix} \in \mathbb{R}^{2N \times 4N}, \quad g_k = \begin{bmatrix} M^{-1}f(t_k) \\ M^{-1}f(t_k + \Delta t/3) \\ M^{-1}f(t_k + 2\Delta t/3) \\ M^{-1}f(t_{k+1}) \end{bmatrix} \in \mathbb{R}^{4N} \]

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

\[ T = \sum_{j=0}^{\infty} t_j(\Delta t) \otimes A^j, \quad L = \sum_{j=0}^{\infty} l_j(\Delta t) \otimes A^j \]

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

\[ t_j(\Delta t) = \frac{(-1)^j \Delta t^{(2j)} - 1}{(2j)!} \begin{bmatrix} \Delta t & \Delta t^2/(2j + 1) \\ 2j & \Delta t \end{bmatrix}, \quad j \geq 0 \]

\[ l_j(\Delta t) = \frac{(-1)^j \Delta t^{(2j)} + 1}{(2j + 4)!} \begin{bmatrix} (j + 1)(8j^2 + 18j + 13)\Delta t & 36(j + 1)^2\Delta t & -9(2j^2 + j - 1)\Delta t & 2(1 + 2j^2)\Delta t \\ 2j + 5 & 2j + 5 & 2j + 5 & 2j + 5 \\ 2j + 5 & 9(2j + 1)^2 & -9(j - 1)(2j + 1) & 4j^2 - 4j + 3 \end{bmatrix} \]

The scheme of Eq. (13) 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. (4), the nth iteration of the asymptotic expansion, \( x^{(n)}(t) \), is in turn solution of the undamped problem with an applied force proportional to the previous iteration. Thus, we can use again the Green’s functions-based approach, yielding

\[ x_{k+1}^{(n)} = G(\Delta t) x_k^{(n)} + H(\Delta t) \ddot{x}_k^{(n)} - \int_{t=t_k}^{t_{k+1}} H(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) dt \]

\[ \ddot{x}_{k+1}^{(n)} = \dot{G}(\Delta t) x_k^{(n)} + \dot{H}(\Delta t) \ddot{x}_k^{(n)} - \int_{t=t_k}^{t_{k+1}} \dot{G}(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) dt \]

where now \( x_k^{(n)} = x^{(n)}(t_k) \) and \( \ddot{x}_k^{(n)} = \ddot{x}^{(n)}(t_k) \), with initial conditions \( x_0^{(n)} = \ddot{x}_0^{(n)} = 0 \). Above, \( \dot{x}^{(n-1)}(t) \) is not explicitly available, since the solution is being determined from the scheme at the samples \( t_k \), \( k = 0, 1, 2, \ldots \). Eqs. (20) will be transformed into an iterative scheme in two stages. First, let us integrate by parts to express the integrals of Eq. (20) in terms of the displacements field \( x^{(n-1)}(t) \). Thus, using the
After some rearrangements, Eqs. (20) lead to the following double iterative scheme, both in velocities at 
\[ \int_{t=t_k}^{t_{k+1}} H(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) \, dt = H(\Delta t) M^{-1} C x_k^{(n-1)} \]
\[ - \int_{t=t_k}^{t_{k+1}} G(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) \, dt \]
\[ - \int_{t=t_k}^{t_{k+1}} G(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) \, dt = -G(0) M^{-1} C x_{k+1}^{(n-1)} + G(\Delta t) M^{-1} C x_k^{(n-1)} + A \int_{t=t_k}^{t_{k+1}} H(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) \, dt \]

The second stage consist of the approximation of \( \dot{x}^{(n-1)}(t) \) 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 \( \xi \) is defined as
\[ \xi = \frac{t - t_k}{t_{k+1} - t_k}, \quad 0 \leq \xi \leq 1, \quad t_k \leq t \leq t_{k+1} \]

The value of \( \dot{x}^{(n-1)}(t) \) can then be approximated by
\[ \dot{x}^{(n-1)}(t) \approx N_1(\xi) x_k^{(n-1)} + D_1(\xi) \Delta t \dot{x}_k^{(n-1)} + N_2(\xi) x_{k+1}^{(n-1)} + D_2(\xi) \Delta t \dot{x}_{k+1}^{(n-1)} \]

where
\[ N_1(\xi) = (1 - \xi)^2(1 + 2\xi), \quad N_2(\xi) = (3 - 2\xi) \xi^2, \quad D_1(\xi) = (1 - \xi)^2 \xi, \quad D_2(\xi) = -(1 - \xi) \xi^2 \]

By means of this interpolation, the function and its derivative coincide with those evaluated from the approximation at the two ends of the interval, at \( t = \{t_k, t_{k+1}\} \). After plugging Eq. (24) into Eq. (23), the integrals can explicitely be written in terms of the \((n-1)\)th iteration at \( t_k \) and \( t_{k+1} \), yielding
\[ - \int_{t=t_k}^{t_{k+1}} H(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) \, dt = \alpha_{uu} x_k^{(n-1)} + \alpha_{uv} \dot{x}_k^{(n-1)} + \beta_{uu} x_{k+1}^{(n-1)} + \beta_{uv} \dot{x}_{k+1}^{(n-1)} \]
\[ - \int_{t=t_k}^{t_{k+1}} G(t_{k+1} - t) M^{-1} C \dot{x}^{(n-1)}(t) \, dt = \alpha_{vu} x_k^{(n-1)} + \alpha_{vv} \dot{x}_k^{(n-1)} + \beta_{vu} x_{k+1}^{(n-1)} + \beta_{vv} \dot{x}_{k+1}^{(n-1)} \]

where the \( N \times N \) matrices of coefficients are
\[ \alpha_{uu} = H(\Delta t) M^{-1} C - \Delta t \int_{\xi=0}^{1} G(\Delta t(1 - \xi)) M^{-1} C N_1(\xi) \, d\xi \]
\[ \alpha_{uv} = G(\Delta t) M^{-1} C + \Delta t A \int_{\xi=0}^{1} H(\Delta t(1 - \xi)) M^{-1} C N_1(\xi) \, d\xi \]
\[ \alpha_{vu} = -\Delta t \int_{\xi=0}^{1} G(\Delta t(1 - \xi)) M^{-1} C N_1(\xi) \, d\xi \]
\[ \alpha_{vv} = \Delta t^2 A \int_{\xi=0}^{1} H(\Delta t(1 - \xi)) M^{-1} C N_1(\xi) \, d\xi \]
\[ \beta_{uu} = -\Delta t \int_{\xi=0}^{1} G(\Delta t(1 - \xi)) M^{-1} C N_2(\xi) \, d\xi \]
\[ \beta_{uv} = -\Delta t \int_{\xi=0}^{1} G(\Delta t(1 - \xi)) M^{-1} C N_2(\xi) \, d\xi \]
\[ \beta_{vu} = -\Delta t \int_{\xi=0}^{1} G(\Delta t(1 - \xi)) M^{-1} C N_2(\xi) \, d\xi \]
\[ \beta_{vv} = \Delta t^2 A \int_{\xi=0}^{1} H(\Delta t(1 - \xi)) M^{-1} C N_2(\xi) \, d\xi \]

After some rearrangements, Eqs. (24) lead to the following double iterative scheme, both in \( k \) and \( n \)
\[ \dot{x}_{k+1}^{(n)} = T \dot{x}_k^{(n)} + \alpha \dot{x}_k^{(n)} + \beta \dot{x}_{k+1}^{(n-1)} \]

\[ n \geq 1, \quad k \geq 0 \]
where the $n$th state vector and the initial conditions are

$$X_k^{(n)} = \left\{ x_k^{(n)}, u_k^{(n)} \right\}, \quad X_0^{(n)} = \left\{ 0, 0 \right\}, \quad n \geq 1$$  \hspace{1cm} (29)

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

$$\alpha = \begin{bmatrix} \alpha_{uu} & \alpha_{uv} \\ \alpha_{vu} & \alpha_{vv} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{uu} & \beta_{uv} \\ \beta_{vu} & \beta_{vv} \end{bmatrix}$$  \hspace{1cm} (30)

Eqs. (29) can analytically be solved using the series expansion of $G(t)$ and $H(t)$ given by Eq. (9). After some algebra, the matrices $\alpha$ and $\beta$ can be written as

$$\alpha = \sum_{j=0}^{\infty} \alpha_j(\Delta t) \otimes (A^j M^{-1} C) = \left[ \sum_{j=0}^{\infty} \alpha_j(\Delta t) \otimes A^j \right] \cdot (I_2 \otimes M^{-1} C)$$

$$\beta = \sum_{j=0}^{\infty} \beta_j(\Delta t) \otimes (A^j M^{-1} C) = \left[ \sum_{j=0}^{\infty} \beta_j(\Delta t) \otimes A^j \right] \cdot (I_2 \otimes M^{-1} C)$$  \hspace{1cm} (31)

where the sequences $\{\alpha_j(\Delta t)\}$ and $\{\beta_j(\Delta t)\}, j \geq 0$, of $2 \times 2$ matrices are defined as

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

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

Eqs. (14) and (23) 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. $X_k^{(n)}$, starting from $X_0^{(0)} = U_0$ and $X_0^{(n)} = 0, n \geq 1$. In the next section the series which gives rise to the asymptotic solution will be summed.

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 $U_k^{(n)}$. Namely,

$$U_k^{(n)} = \left\{ u_k^{(n)}(t_k), \dot{u}_k^{(n)}(t_k) \right\} \equiv \left\{ u_k^{(n)}, \dot{u}_k^{(n)} \right\} = X_k^{(0)} + X_k^{(1)} + \cdots + X_k^{(n)} = \sum_{\nu=0}^{n} X_k^{(\nu)}$$  \hspace{1cm} (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

$$U_k = \left\{ u_k, \dot{u}_k \right\} \equiv \lim_{n \to \infty} U_k^{(n)} = \sum_{n=0}^{\infty} X_k^{(n)}, \quad k \geq 0$$  \hspace{1cm} (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, $0 \leq \nu \leq n$

$$X_k^{(0)} = T X_k^{(0)} + L g_k, \quad \nu = 0$$

$$-\beta X_k^{(0)} + X_k^{(1)} = T X_k^{(1)} + \alpha X_k^{(0)}, \quad \nu = 1$$

$$\vdots$$

$$-\beta X_k^{(n-1)} + X_k^{(n)} = T X_k^{(n)} + \alpha X_k^{(n-1)}, \quad \nu = n$$  \hspace{1cm} (36)
Summing the left and the right side terms of the above equations and taking into account that \( U^{(n)}_k = \sum_{\nu=0}^{\infty} X^{(\nu)}_k \), yields

\[ -\beta U^{(n-1)}_{k+1} + U^{(n)}_k = T U^{(n)}_k + \alpha U^{(n-1)}_k + L g_k, \quad n \geq 1 \]  

(37)

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

\[ (I_{2N} - \beta) U_{k+1} = (T + \alpha) U_k + L g_k \]  

(38)

which, however, can be written in explicit form just multiplying by the corresponding inverse matrix, namely

\[ U_{k+1} = a U_k + b_k, \quad k = 0, 1, 2, \ldots \]  

(39)

where the main matrices of the algorithm, \( a \in \mathbb{R}^{2N \times 2N} \) and \( b_k \in \mathbb{R}^{2N} \), are

\[ a = (I_{2N} - \beta)^{-1} (T + \alpha) \in \mathbb{R}^{2N \times 2N} \]  

(40)

\[ b_k = (I_{2N} - \beta)^{-1} L g_k \in \mathbb{R}^{2N}, \quad k = 0, 1, 2, \ldots \]  

(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 article. In the following sections, the properties of the numerical approach will be discussed, indeed

Sec. 4: Conditions for convergence of asymptotic expansion \( U_k = \sum_{n=0}^{\infty} X^{(n)}_k \)

Sec. 5: Algorithm for the computation of main matrices, \( a \) and \( b_k \).

Sec. 6: Analysis of stability

Sec. 7: Order of accuracy

Sec. 8: Computation effort

4. Analysis of convergence

The 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 is convergent. In this section, the conditions under which such limit actually exists will be addressed. 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. Evaluations of Eq. (28) for \( n \geq 2 \) and for \( k = \{0, 1, \ldots, k - 1\} \) yields

\[ X^{(n)}_1 = T X^{(n)}_0 + \alpha X^{(n-1)}_0 + \beta X^{(n-1)}_1 \]

\[ X^{(n)}_2 = T X^{(n)}_1 + \alpha X^{(n-1)}_1 + \beta X^{(n-1)}_2 \]

\[ \vdots \]

\[ X^{(n)}_k = T X^{(n)}_{k-1} + \alpha X^{(n-1)}_{k-1} + \beta X^{(n-1)}_k, \quad n \geq 2 \]  

(42)

Since \( X^{(n)}_0 = X^{(n-1)}_0 = 0 \) for \( n \geq 2 \), then the first step \( (k = 0) \) is reduced to \( X^{(n)}_1 = \beta X^{(n-1)}_1 \). The terms \( X^{(n)}_1, \ldots, X^{(n)}_{k-1} \) 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, Eqs. (42)
can be written separating the terms associated to the \( n \)th and \( (n-1) \)th iterations in both sides of equations, i.e.

\[
\begin{align*}
X_1^{(n)} &= \beta X_1^{(n-1)} \\
X_2^{(n)} &= \gamma X_1^{(n-1)} + \beta X_2^{(n-1)} \\
X_3^{(n)} &= T\gamma X_1^{(n-1)} + \beta X_3^{(n-1)} \\
& \vdots \\
X_k^{(n)} &= \sum_{r=1}^{k-1} T^{k-r} \gamma X_{r}^{(n-1)} + \beta X_k^{(n-1)}, \quad n \geq 2
\end{align*}
\]

(43)

where \( \gamma = T\beta + \alpha \). Rearranging \( X_1^{(n)}, \ldots, X_k^{(n)} \) in one single \( 2Nk \times 2Nk \)–size column vector denoted by \( X^{(n)} \in \mathbb{R}^{2Nk} \), and after some algebra, the above relations can be expressed as

\[
X^{(n)} = RX^{(n-1)}, \quad n \geq 2
\]

(44)

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

\[
X^{(n)} = \begin{bmatrix} X_1^{(n)} \\ \vdots \\ X_k^{(n)} \end{bmatrix} \in \mathbb{R}^{2Nk}, \quad R = \begin{bmatrix} \beta & 0_{2N} & 0_{2N} & \cdots & 0_{2N} & 0_{2N} \\ \gamma & \beta & 0_{2N} & \cdots & 0_{2N} & 0_{2N} \\ T\gamma & \gamma & \beta & \cdots & 0_{2N} & 0_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ T^{k-3} \gamma & T^{k-4} \gamma & T^{k-5} \gamma & \cdots & \beta & 0_{2N} \\ T^{k-2} \gamma & T^{k-3} \gamma & T^{k-4} \gamma & \cdots & \gamma & \beta \end{bmatrix} \in \mathbb{R}^{2Nk \times 2Nk}
\]

(45)

The first iteration for \( n = 1 \) leads to the relationship between \( X^{(0)} \) and \( X^{(1)} \). It can be derived straightforward when Eq. (43) is evaluated and taking into account that \( X_0^{(1)} = 0 \) and \( X_0^{(0)} = U_0 \). As before, using some matrix algebra, the following expression can be obtained

\[
X^{(1)} = RX^{(0)} + SU_0
\]

(46)

where

\[
S = \begin{bmatrix} \alpha \\ T\alpha \\ \vdots \\ T^{k-1} \alpha \end{bmatrix} \in \mathbb{R}^{2Nk \times 2N}
\]

(47)

We can build the \( 2Nk \times 2Nk \)–size column vector \( U \) with the response of all time steps, resulting the series expansion

\[
U = \begin{bmatrix} U_1 \\ \vdots \\ U_k \end{bmatrix} = \begin{bmatrix} \sum_{n=0}^{\infty} X_1^{(n)} \\ \vdots \\ \sum_{n=0}^{\infty} X_k^{(n)} \end{bmatrix} = \sum_{n=0}^{\infty} \begin{bmatrix} X_1^{(n)} \\ \vdots \\ X_k^{(n)} \end{bmatrix} = \sum_{n=0}^{\infty} X^{(n)}
\]

(48)

Using the recursive relations of Eq. (44) for \( n \geq 2 \) together with the first iteration \((n = 1)\) from Eq. (46), the response \( U \) results

\[
U = X^{(0)} + X^{(1)} + X^{(2)} + \cdots + X^{(n)} + \cdots = X^{(0)} + \sum_{n=0}^{\infty} R^n X^{(1)} = \left( \sum_{n=0}^{\infty} R^n \right) \left( X^{(0)} + SU_0 \right)
\]

(49)

This closed-form relationship highlights the intrinsic nature of the proposed approach as an asymptotic expansion-based method. Note that both matrices \( R \) and \( S \) are directly proportional to the damping
matrices, so that for undamped problems we have \( U \equiv X^{(0)} \). The terms arranged within \( X^{(n)} \) represent the \( n \)th order contribution to the response and their effect is expected to decrease significantly for low damping problems. The proposed method leads to a solution as long as the Neumann series converges. The necessary and sufficient condition for this series to be convergent is that all the eigenvalues of matrix \( R \) are strictly smaller than the unit in absolute value [49]. Therefore, the series is convergent provided that \( \rho(R) < 1 \), where \( \rho(\bullet) \) denotes the spectral radius of a matrix. Since \( R \) is a triangular matrix by blocks and \( \beta \) is the matrix repeated in the block diagonal, then eigenvalues of \( R \) are those of \( \beta \), and the convergence of the method holds provided that

\[
\rho(\beta) < 1
\]  

(50)

The matrix \( \beta \) has already been presented in Eqs. [31] and [33] as a power series expansion of the system matrix \( A = M^{-1}K \). Moreover, in view of the form of the matrix all terms are proportional to the dissipative forces. Analytically, the expression of \( \beta \) given above can be expressed as a product of two matrices

\[
\beta = \Delta t \left[ \lim_{m \to \infty} \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) \right] \cdot (I_2 \otimes M^{-1}C)
\]

(51)

where

\[
\sigma_m(\tau) = \sum_{j=0}^{m/2} \frac{(-1)^j}{(2j+4)!} \begin{bmatrix}
-12(\tau^2j + 1) & 2(2j+1)\Delta t \tau^2j \\
-12(2j+1)(j+2)\tau^2j / \Delta t & 8j(j+2)\tau^2j
\end{bmatrix}
\]

(52)

The matrix \( \sigma_m(\tau) \) has 2 \times 2 elements as long as the argument \( \tau \in \mathbb{R} \), but it will be \( \sigma(\tau) \in \mathbb{R}^{2N \times 2N} \) if \( \tau \in \mathbb{R}^{N \times N} \), as shown in Eq. [51] with \( \tau = \Delta t \sqrt{M^{-1}K} \). The index \( m \) denotes the maximum order of \( \tau \) in the power series expansion. In practice, a finite value for \( m \) means the truncation of the series.

The nature of the matrix \( \beta \) is revealed in the decomposition shown in Eq. [51]. Both time step \( \Delta t \) and damping matrix \( C \) play a relevant role in its spectral radius. Let us see that, under certain conditions, some practical outcomes can be drawn to estimate a priori an upper bound of the spectral radius \( \rho(\beta) \) as a function of \( \Delta t \) and of the system matrices. Indeed, it is known that the spectral radius of a product of matrices is less than or equal to the product of the respective spectral radii whenever the matrices involved commute. In the particular case of Eq. [51], the matrices \( \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) \) and \( (I_2 \otimes M^{-1}C) \) commute provided that \( M^{-1}K \) and \( M^{-1}C \) commute, something that holds for classically (or proportionally) damped systems [54]. In such a case, the spectral radius is bounded by

\[
\rho(\beta) \leq \rho \left[ \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) \right] \Delta t \rho \left( M^{-1}C \right)
\]

(53)

The four blocks of the matrix \( \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) \) can be diagonalized in the modal space of the undamped problem, allowing the analytical determination of the eigenvalues. By modal analysis, the undamped natural frequencies and eigenvectors \( \omega_j \in \mathbb{R} \) and \( \phi_j \in \mathbb{R}^N \), are related throughout

\[
K \phi_j = \omega_j^2 M \phi_j, \quad 1 \leq j \leq N
\]

(54)

Assuming that there are not repeated frequencies, the eigenvectors can be mass-normalized using the biorthogonality relations \( \phi_j^T M \phi_k = \delta_{jk} \) and \( \phi_j^T K \phi_k = \omega_j^2 \delta_{jk} \) where \( \delta_{jk} \) is the Kronecker delta. The eigenvectors \( \phi_j \) can be arranged as the columns of the square matrix (modal matrix) \( \Phi = [\phi_1, \ldots, \phi_N] \in \mathbb{R}^{N \times N} \) so that the biorthogonality relations can be written in matrix form as \( \Phi^T M \Phi = I_N, \phi_j^T K \phi_k = \Omega^2 \) where \( I_N \in \mathbb{R}^{N \times N} \) denotes the identity matrix and \( \Omega = \text{diag}[\omega_1, \ldots, \omega_N] \) is a diagonal matrix with the natural frequencies. From these relations it is verified that

\[
M^{-1}K = \Phi \Omega^2 \Phi^{-1}, \quad \sqrt{M^{-1}K} = \Phi \Omega \Phi^{-1}
\]

(55)
and therefore
\[ \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) = (I_2 \otimes \Phi) \cdot \sigma_m (\Delta t \Omega) \cdot (I_2 \otimes \Phi^{-1}) \] (56)

From this matrix relation, both \( \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) \) and \( \sigma_m (\Delta t \Omega) \) are similar matrices and consequently they share the same eigenvalues. In addition, the computation of eigenvalues from matrix \( \sigma_m (\Delta t \Omega) \) is straightforward because it is formed by four square diagonal blocks. The set of \( 2N \) eigenvalues yields
\[ \bigcup_{i=1}^{N} \{ \mu_1(m, \omega_i \Delta t), \mu_2(m, \omega_i \Delta t) \} \] (57)

where \( \{ \mu_1(m, \tau), \mu_2(m, \tau) \} \) denotes the two eigenvalues of the \( 2 \times 2 \) matrix \( \sigma_m (\tau) \) when \( \tau \in \mathbb{R} \), which can analytically be computed. Thus, for instance for \( m = 0 \) and \( m = 2 \) the results are

\[ \sigma_0(\tau) = \begin{bmatrix} -1/2 & \Delta t/12 \\ -1/\Delta t & 0 \end{bmatrix} \quad \mu_{1,2}(0, \tau) = \frac{1}{12} \left( -3 \pm i \sqrt{3} \right) \]

\[ \sigma_2(\tau) = \begin{bmatrix} -1/2 + \tau^2/30 & (1/12 - \tau^2/120)\Delta t \\ (1 + 3\tau^2/20) / \Delta t & -\tau^2/30 \end{bmatrix} \quad \mu_{1,2}(2, \tau) = \frac{1}{120} \left( -30 \pm i \sqrt{300 + 2\tau^4 - 60\tau^2} \right) \] (58)

As the order \( m \) increases, the order of \( \tau \) within \( \mu_1(m, \tau) \) and \( \mu_2(m, \tau) \) will increase as well. In general, the spectral radius of matrix \( \sigma_m (\Delta t \sqrt{M^{-1}K}) \) can be written as
\[ \rho \left[ \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) \right] = \max_{1 \leq i \leq N} \rho [\sigma_m (\omega_i \Delta t)] \] (59)

Substituting the above expression into Eq. (53), then the general condition for the proposed numerical scheme in classically damped systems to be convergent is
\[ \rho(\beta) \leq \Delta t \rho (M^{-1}C) \max_{1 \leq i \leq N} \rho [\sigma_m (\omega_i \Delta t)] < 1 \] (60)

The distribution of eigenvalues (in absolute value) of \( \sigma_m (\tau) \) is shown in fig. 1 for different values of the truncation order \( m \). For \( m = 0 \), the eigenvalues of \( \sigma_0(\tau) \) are independent of \( \tau \) and the spectral radius is \( \rho[\sigma_0] = 1/2 \sqrt{3} \). On the other side, the eigenvalues of \( \sigma_\infty(\tau) \), as function of \( \tau \), draw a decreasing curve as \( \tau \) increases, as shown in fig. 1 (solid black line). Assuming infinite terms, i.e. \( m = \infty \), it can be ensured that \( \rho[\sigma_\infty(\tau)] \leq \rho(\sigma_0) = 1/2 \sqrt{3} \) and therefore \( \rho(\beta) < 1 \) provided that
\[ \Delta t \rho (M^{-1}C) < 2 \sqrt{3} , \quad (m = \infty) \] (61)
In practice, the order of truncation \( m \) must be finite. Hence, for each \( m < \infty \), convergence does not hold within the whole range of \( \omega_i \Delta t \), existing a limit value of \( \tau \) from which Eq. (63) is unbounded. Let us denote by \( \tau_L(m) \) to the solution of the equation \( \rho [\sigma_m(\tau)] = 1/2\sqrt{3} \). In Fig. 1 the values of \( \tau_L(m) \) have been highlighted for \( m = 2, 10 \) and 20 using a square, circle and diamond marker, respectively. Then, for each value of \( m \), it is verified that

\[
\rho \left[ \sigma_m \left( \Delta t \sqrt{M^{-1}K} \right) \right] = \max_{i \leq i \leq N} \rho [\sigma_m(\omega_i \Delta t)] \leq \frac{1}{2\sqrt{3}} , \quad 0 \leq \omega_i \Delta t < \tau_L(m)
\]  

(62)

The inequalities \( \omega_i \Delta t < \tau_L(m) \), \( 1 \leq i \leq N \) hold provided that it holds for the last frequency, i.e. \( \omega_{\text{max}} \Delta t < \tau_L(m) \), where \( \omega_{\text{max}} = \rho(\sqrt{M^{-1}K}) = 2\pi/T \) stands for the maximum natural frequency in the system (rad/s) and \( T \) for the minimum period (in seconds). Substituting Eq. (62) into Eq. (60), the condition \( \rho(\beta) < 1 \) can be written as

\[
\rho(\beta) \leq \Delta t \rho \left( M^{-1}C \right) \max_{1 \leq i \leq N} \rho [\sigma_m(\omega_i \Delta t)] \leq \Delta t \rho \left( M^{-1}C \right) \frac{1}{2\sqrt{3}} < 1 , \quad 0 \leq \omega_{\text{max}} \Delta t < \tau_L(m)
\]  

(63)

Writing together both restrictions of Eq. (63) and fixed a truncation order \( m \), it can be concluded that for classically damped systems, \( \rho(\beta) < 1 \) provided that both \( \Delta t \rho(M^{-1}C) < 2\sqrt{3} \) and \( \omega_{\text{max}} \Delta t < \tau_L(m) \) hold simultaneously. The following condition encapsulates both restrictions in a single expression as a limitation of the time step,

\[
\Delta t \leq \min \left\{ \frac{2\sqrt{3}}{\rho(M^{-1}C)}, \frac{\tau_L(m)}{\rho(\sqrt{M^{-1}K})} \right\} = \min \left\{ \frac{2\sqrt{3}}{\rho(M^{-1}C)}, \frac{\tau_L(m)T}{2\pi} \right\}
\]  

(64)

where the limit values \( \tau_L(m) \) are shown in Table 1 for some truncation orders. Although the condition derived in Eq. (64) is strictly valid for classically damped systems, it can be considered acceptable for systems with light damping, which can be assumed to be slightly non-proportional, allowing a priori an estimation of the time step \( \Delta t \). Furthermore, Eq. (64) provides information about the constraints on the application of the numerical model. Thus, in the presence of weak damping, the condition \( \Delta t/T < \tau_L(m)/2\pi \) will be limiting. On the other hand, highly damped systems can also be analyzed with the proposed method, provided that the time step is sufficiently small.

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 \( A(\Delta t) \) and the vector \( b_k(\Delta t) \). Throughout

| Truncation order \( m \) | \( \tau_L(m) \) | \( \tau_L(m)/2\pi \) |
|--------------------------|-------------|------------------|
| \( m = 2 \)             | 2.64303     | 0.42065          |
| \( m = 4 \)             | 9.59335     | 1.52683          |
| \( m = 6 \)             | 5.48854     | 0.87353          |
| \( m = 8 \)             | 6.68027     | 1.06320          |
| \( m = 10 \)            | 7.38332     | 1.17509          |
| \( m = 20 \)            | 11.3105     | 1.80012          |
| \( m = 30 \)            | 15.1700     | 2.41438          |
| \( m = 40 \)            | 19.0203     | 3.02718          |

Table 1: Values of \( \tau_L(m) \) for different truncation orders, \( m \). \( \tau_L(m) \) is the solution of equation \( \rho [\sigma_m(\tau)] = 1/2\sqrt{3} \). Once the truncation order has been set, the parameter \( \tau_L(m) \) is used to calculate the maximum time step admissible by the numerical method, for convergence reasons, being \( \Delta t \leq \{2\sqrt{3}/\rho(M^{-1}C), \tau_L(m)T/2\pi\} \).
The iterative process consists of steps, namely

\[
\begin{align*}
\delta a(\Delta t) &= 2 \delta a(\Delta t/2^p) + \delta a(\Delta t/2^{p-1}) \cdot \delta a(\Delta t/2^p) \\
\delta a(\Delta t/2^{p-2}) &= 2 \delta a(\Delta t/2^{p-1}) + \delta a(\Delta t/2^{p-1}) \cdot \delta a(\Delta t/2^{p-1}) \\
\vdots \quad (p \text{ times}) \quad \vdots \\
\delta a(\Delta t_0) &= 2 \delta a(\Delta t/2) + \delta a(\Delta t/2) \cdot \delta a(\Delta t/2)
\end{align*}
\]

Consider then the reduced time-step introduced above, i.e. \( \Delta t_0 = \Delta t/2^p \), the first step is to compute \( \delta a(\Delta t_0) \). Note that this latter is very small with \( \| \delta a(\Delta t_0) \| \ll 1 \), so that using the iterative process of Eq. (74) results much more effective from a round-off error perspective than using the total matrices \( a(\Delta t) \). The iterative process consists of \( p \) steps, namely

\[
\begin{align*}
\delta a(\Delta t/2^{p-1}) &= 2 \delta a(\Delta t/2^p) + \delta a(\Delta t/2^p) \cdot \delta a(\Delta t/2^p) \\
\delta a(\Delta t/2^{p-2}) &= 2 \delta a(\Delta t/2^{p-1}) + \delta a(\Delta t/2^{p-1}) \cdot \delta a(\Delta t/2^{p-1}) \\
\vdots \quad (p \text{ times}) \quad \vdots \\
\delta a(\Delta t) &= 2 \delta a(\Delta t/2) + \delta a(\Delta t/2) \cdot \delta a(\Delta t/2)
\end{align*}
\]

this section it is necessary to highlight the dependence of the parameter \( \Delta t \) in both matrices. Rewriting their expressions

\[
a(\Delta t) = (I_{2N} - \beta)^{-1} (T + \alpha) \tag{65}
b_k(\Delta t) = (I_{2N} - \beta)^{-1} L g_k, \quad k = 0, 1, 2, \ldots \tag{66}
\]

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

\[
(I_{2N} - \beta)^{-1} = \sum_{r=0}^{\infty} \beta^r \approx I_{2N} + \beta + \beta^2 + \cdots + \beta^r \tag{67}
\]

In the following points the algorithms for the efficient computation of both \( a(\Delta t) \) and \( b_k(\Delta t) \) will be described in detail. Although the same matrix \( \beta \) arises in both Eqs. (65) and (66), 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 \( a(\Delta t) \)

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

\[
U_{k+1} = a(\Delta t) U_k \tag{68}
\]

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

\[
a(\Delta t) = \left[ a \left( \frac{\Delta t}{2^p} \right) \right]^{2^p} = [a(\Delta t_0)]^{2^p} \tag{69}
\]

Moreover, \( a(\Delta t_0) \) can be expressed in incremental form as

\[
a(\Delta t_0) = I_{2N} + \delta a(\Delta t_0) \tag{70}
\]

where \( \delta a(\Delta t_0) \) is a very small matrix. Strictly it can be stated that \( \lim_{\Delta t \to 0} \delta a(\Delta t) = 0 \). The relationship of Eq. (69) allows the use of the so-called \( 2^p \) algorithm, commonly used for the evaluation of exponential matrices [51]. Moreover, Eq. (70) enables to use this technique on the incremental part \( \delta a(\Delta t) \) instead of on the total matrix, improving significantly the accuracy. Thus, the product \( a(\Delta t) a(\Delta t/2) = a(\Delta t) = I_{2N} + \delta a(\Delta t) \) produces

\[
\delta a(\Delta t) = 2 \delta a(\Delta t_0) + \delta a(\Delta t_0) \cdot \delta a(\Delta t_2) \cdot \delta a(\Delta t_2) \tag{71}
\]

Consider then the reduced time-step introduced above, i.e. \( \Delta t_0 = \Delta t/2^p \), the first step is to compute \( \delta a(\Delta t_0) \). Note that this latter is very small with \( \| \delta a(\Delta t_0) \| \ll 1 \), so that using the iterative process of Eq. (71) results much more effective from a round-off error perspective than using the total matrices \( a(\Delta t) \). The iterative process consists of \( p \) steps, namely

\[
\begin{align*}
\delta a(\Delta t/2^{p-1}) &= 2 \delta a(\Delta t/2^p) + \delta a(\Delta t/2^p) \cdot \delta a(\Delta t/2^p) \\
\delta a(\Delta t/2^{p-2}) &= 2 \delta a(\Delta t/2^{p-1}) + \delta a(\Delta t/2^{p-1}) \cdot \delta a(\Delta t/2^{p-1}) \\
\vdots \quad (p \text{ times}) \quad \vdots \\
\delta a(\Delta t_0) &= 2 \delta a(\Delta t/2) + \delta a(\Delta t/2) \cdot \delta a(\Delta t/2)
\end{align*}
\]
and finally the main matrix of the algorithm is determined as $a(\Delta t) = I_{2N} + \delta a(\Delta t)$. 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 $\delta a(\Delta t_0) = a(\Delta t_0) - I_{2N}$. As Eq. (65) shows, the computation of $a(\Delta t_0)$ requires the truncation of two series: (i) those of $T$, $\alpha$ and $\beta$ and (ii) the Neumann series corresponding to $(I_{2N} - \beta)^{-1}$. This lead us to introduce the following two parameters

(i) The parameter $m_a$ stands for the truncation order of matrices $T$, $\alpha$, $\beta$. 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, given an even integer $m_a$ we introduce the following approximations

$$T \approx I_{2N} + \delta T_a(\Delta t_0) = I_{2N} + \left[ \begin{array}{cc} \delta G_a(\Delta t_0) & H_a(\Delta t_0) \\ -A H_a(\Delta t_0) & \delta G_a(\Delta t_0) \end{array} \right] \quad (73)$$

$$\delta G_a(\Delta t_0) = \sum_{j=1}^{m_a/2} (-1)^j \left( \frac{1}{2j} \right)^2 (\Delta t_0^2 A)^j \quad , \quad H_a(\Delta t_0) = \Delta t_0 \sum_{j=0}^{m_a/2} \frac{(-1)^j}{(2j+1)!} (\Delta t_0^2 A)^j \quad (74)$$

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

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

$$(I_{2N} - \beta_a)^{-1} \approx I_{2N} + \beta_a + \beta_a^2 + \cdots + \beta_a^{r_a} \equiv I_{2N} + \delta \beta_a \quad (75)$$

where the matrix $\delta \beta_a = \beta_a + \beta_a^2 + \cdots + \beta_a^{r_a}$.

Substituting Eqs. (73), (74) and (75) into Eq. (65), and after some matrix products we have

$$a(\Delta t_0) = (I_{2N} - \beta)^{-1} (T + \alpha) \approx (I_{2N} + \delta \beta_a) (I_{2N} + \delta T_a + \alpha_a)$$

$$= I_{2N} + (\delta T_a + \alpha_a + \delta \beta_a + \delta \beta_a \delta T_a + \delta \beta_a \alpha_a) \equiv I_{2N} + \delta a(\Delta t_0) \quad (76)$$

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

$$\delta a(\Delta t_0) = \delta T_a + \alpha_a + \delta \beta_a + \delta \beta_a \delta T_a + \delta \beta_a \alpha_a \quad (77)$$

Taking $p = 20$, then $\Delta t_0 = \Delta t / 1048576$ is a very small quantity so that it is sufficient to take $m_a = 2$ to achieve a highly precise estimation of $\delta a(\Delta t_0)$. Furthermore, as will be shown later, the case $m_a = 2$ behaves very favorably respect to the stability for any value of the damping forces, even the undamped case.
The form of the matrices is transcribed below for this particular case \((m_a = 2)\).

\[
\delta T_a = \begin{bmatrix}
-\Delta t_0 A + \frac{\Delta t_0^2}{6} A^2 & \Delta t_0 I_N - \frac{\Delta t_0^3}{2} A \\
-\Delta t_0 A & \Delta t_0 I_N - \frac{\Delta t_0^3}{2} A
\end{bmatrix}
\]  

(78)

\[
\alpha_a = \begin{bmatrix}
\frac{\Delta t_0^2}{2} M^{-1} C - \frac{\Delta t_0^3}{20} A M^{-1} C - \frac{\Delta t_0^4}{12} M^{-1} C + \frac{\Delta t_0^5}{20} A M^{-1} C \\
M^{-1} C - \frac{\Delta t_0^2}{20} A M^{-1} C
\end{bmatrix}
\]  

(79)

\[
\beta_a = \begin{bmatrix}
-\frac{\Delta t_0^2}{2} M^{-1} C + \frac{\Delta t_0^3}{20} A M^{-1} C + \frac{\Delta t_0^4}{12} M^{-1} C - \frac{\Delta t_0^5}{20} A M^{-1} C \\
-\frac{\Delta t_0^2}{20} A M^{-1} C
\end{bmatrix}
\]  

(80)

\[
\delta \beta_a = \beta_a + \beta_a^2 + \ldots + \beta_a^q
\]  

(81)

The Algorithm 1 outlines the steps necessary to obtain the matrix \(a(\Delta t)\). 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.

Algorithm 1 Computation of main matrix \(a(\Delta t)\)

1: Fix main parameters, \(\Delta t, r_a = 2, m_a = 2\)
2: Compute matrices \(M^{-1} C, A = M^{-1} K\)
3: Evaluate \(\Delta t_0 = \Delta t/2^p\) \((p = 20)\)
4: Compute matrix \(\delta T_a\), Eq. \(78\)
5: Compute matrix \(\alpha_a\), Eq. \(79\)
6: Compute matrix \(\beta_a\), Eq. \(80\)
7: Compute matrix \(\delta \beta_a = \beta_a + \beta_a^2\), Eq. \(81\)
8: Compute matrix \(\delta a(\Delta t_0) = \delta T_a + \alpha_a + \delta \beta_a + \delta \beta_a \delta T_a + \delta \beta_a \alpha_a\), Eq. \(77\)
9: Initialize \(\delta a := \delta a(\Delta t_0)\)
10: For \(j = 1 \ldots q\) do
11: \(\delta a = 2\delta a + \delta a \cdot \delta a\), Eqs. \(82\)
12: end for
13: Update \(\delta a(\Delta t) = \delta a\)
14: Evaluate \(a(\Delta t) = I_{2N} + \delta a(\Delta t)\)

5.2. Computation of vector \(b_k(\Delta t)\)

The vector \(b_k(\Delta t)\) is directly related to the nonhomogeneous terms of the transient problem. As shown in Eq. \(66\), it depends on the inverse matrix \((I_{2N} - \beta)^{-1}\) and on the matrix \(L\) defined in turn in terms of a matrix series, as shown in Eqs. \(17\) and \(19\). As for the main matrix \(a(\Delta t)\) studied above, the two series expansions in \(b_k\) 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 \(\beta\) and \(L\). In more detail we have, only with respect to the calculation of \(b_k\)

\[
b_k = (I_{2N} - \beta)^{-1} L g_k \approx (I_{2N} - \beta_b)^{-1} L_b g_k \quad , \quad k = 0, 1, 2, \ldots
\]  

(82)

where

\[
L_b = \sum_{j=0}^{m_b/2} \beta_j(\Delta t) \otimes (A^j M^{-1} C)
\]  

(83)

with \(L_j(\Delta t)\) and \(\beta_j(\Delta t)\), \(2 \times 2\) matrices defined in Eqs. \(19\) and \(83\), respectively. It is important to highlight that both \(L_b\) and \(\beta_b\) are computed using the time step of the problem \(\Delta t\) and not the reduced one, \(\Delta t_0\). As before, the parameter \(m_b\) should be an even number because \(m_b/2\) denotes the number of terms to be taken.
particular value are recommended here. Later in the numerical examples, a study of the sensitivity of
and
r
\( r \)
and in the experiments, the parameters
m
\( m \)
and
a
\( a \)
parameters will be carried out later. To ensure that the above sum converges, we know that the spectral radius of \( \beta_b \) must verify \( \rho(\beta_b) < 1 \). We have already checked in Fig. 1 that the higher
m
\( m \)
, the wider the range of \( \Delta t \) within the eigenvalues of \( \beta_b \) are bounded. In general, increasing the parameter
m
\( m \) will increase the accuracy of the results and decrease the spectral radius of \( \beta_b \), however it will increase the computational cost. A more detailed study of the latter as a function of these parameters will be carried out later.

In the Algorithm 2 the necessary steps to obtain the matrix \( b_k(\Delta t) \) are listed. As shown by the numerical experiments, the parameters
m
\( m \) and
r
\( r \) 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
\( m \) and
r
\( r \) will be carried out.

| Algorithm 2 Computation of main matrix \( b_k(\Delta t) \) |
|---|
| 1: Fix main parameters, \( \Delta t \) and \( r_b, m_b \) (even numbers) |
| 2: Compute matrices \( M^{-1}C, A = M^{-1}K \) |
| 3: Compute and store matrices \( A^j, j = 2, \ldots, m_b/2 \) |
| 4: Compute and store matrices \( A_j(M^{-1}C), j = 2, \ldots, m_b/2 \) |
| 5: Compute sequence of matrices \( I_j(\Delta t) \in \mathbb{R}^{2 \times 4}, 1 \leq j \leq m_b/2 \), Eq. 19 |
| 6: Compute sequence of matrices \( \beta_j(\Delta t) \in \mathbb{R}^{2 \times 2}, 1 \leq j \leq m_b/2 \), Eq. 33 |
| 7: Compute matrix \( L_b \), Eq. 53 |
| 8: Compute matrix \( B \), Eq. 54 |
| 9: Compute matrix \( B = I_{2N} + \beta_b + \beta^2_b \) |
| 10: if \( r_b \geq 4 \) then |
| 11: for \( n = 2 \ldots r_b/2 \) do |
| 12: \( B = I_{2N} + \beta_b + \beta^2_b B \) |
| 13: end for |
| 14: end if |
| 15: Compute vector \( g_b \), Eq. 10 |
| 16: Compute \( b_k(\Delta t) = B L_b g_b \), Eq. 54 |

6. Algorithm stability

The proposed scheme

\[
U_{k+1} = a(\Delta t) U_k + b_k(\Delta t), \quad k = 0, 1, 2, \ldots
\]

is stable provided that

\[
\rho [a(\Delta t)] \leq 1
\]

To discuss the stability of the method it is sufficient to study the single dof oscillator of mass \( M \), spring rigidity \( K \) and damping coefficient \( c = 2M \omega \zeta \). Where \( \omega = \sqrt{K/M} = [\omega^2] \) for the undamped natural frequency, \( T \) for the natural period and \( \zeta \) is the damping ratio. The matrices of the system are then the numbers \( A = [K/M] = [\omega^2] \) and \( M^{-1}C = [\omega^2 \zeta] \). Since the matrix \( a(\Delta t) \) 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
where the truncated matrices, denoted with the subscript \( a \), results

\[
\begin{align*}
T_a &= \sum_{j=0}^{m_a/2} (-1)^j \frac{2j+1}{(2j+1)!} \left[ 2 \Delta t_0 - \tau^2/\Delta t_0 \right] \\
\alpha_a &= \sum_{j=0}^{m_a/2} (-1)^j \frac{24(j+1)}{(2j+1)!} \left[ -4(2j+1)\Delta t_0 - 8(2j+1)(j+1) \right] \\
\beta_a &= \sum_{j=0}^{m_a/2} (-1)^j \frac{24(2j+1)(j+2)\Delta t_0}{(2j+1)!} \left[ -24(2j+1)(j+2)/\Delta t_0 + 4(2j+1)\Delta t_0 \right]
\end{align*}
\]  

(89)

The variable \( \tau = \Delta t_0 \omega = 2\pi \Delta t_0/T \) represents a dimensionless measure of the reduced time step. Eigenvalues \( \lambda \) of matrix \( a(\Delta t_0) \) can be determined as function of the nondimensional time step \( \Delta t_0/T = \tau/2\pi \). As shown in Eqs. (89), \( |\lambda| \) depends also on the parameters \( m_a, r_a \) and on the damping ratio \( \zeta \). Since the reduced time step necessary to compute \( a(\Delta t_0) \) is a very small number, i.e. \( \Delta t_0 = \Delta t/2^p \), with \( p = 20 \), then in general the inequality \( \rho(\beta_a) \ll 1 \) holds and only a few terms in the Neumann series are necessary. In fact, considering \( r_a = 2 \) leads to very accurate results in the majority of cases, something that is supported by the numerical experiments. Figs. (2a), (2b), (2c) and (2d) show the absolute value of eigenvalues of matrix \( a(\Delta t_0) \) 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 2 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 \( \Delta t_0 \). 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 \( \Delta t \) used to guarantee a minimum of accuracy in real problems. As Figs. 2 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 2). The case \( m_a = 2 \) behaves very favorably, showing conditional stability in an interval that increases slightly as the damping increases.

In figs. (2e) and (2f), the absolute value of eigenvalues of matrix \( \beta_a \) are plotted for two cases of damping \( \zeta = 0.05 \) and \( \zeta = 0.5 \). Since \( \beta_a \) is directly proportional to \( \zeta \), the spectral radius decreases also proportionally to \( \zeta \). For our approach to be convergent, we need the spectral radius of the \( \beta_a \) matrix to be less than unity. The convergence interval, with respect to the computation of \( a(\Delta t_0) \), can be specified as the range of values of \( \Delta t_0/T \) for which \( \rho(\beta_a) < 1 \) is verified. Hence, highly damped structures could have an interval of convergence narrower than the interval of stability, in terms of \( \Delta t_0/T \). However, due to the very small size of the reduced time step, this point is not relevant and convergence will still be guaranteed for most practical cases.

7. Order of accuracy

The order of accuracy of an iterative time-step numerical method is the minimum order of \( \Delta t \) in the error term. Denoting by \( U(t_k) \) the \( 2N \)-size array with the exact displacements and velocities at \( t = t_k \) and by \( U_k \) to the proposed numerical approach, then the order of accuracy is said to be \( h \) if

\[
U(t_k) = U_k + \mathcal{O}(\Delta t^h)
\]  

(90)
where $\mathcal{O}(\Delta t^k)$ represents a $2N$ vector expressed as an infinite series in terms of $\Delta t$ whose minimum exponent is precisely $h$. Throughout this section the notation $\mathcal{O}(\bullet)$ will be used interchangeably to refer the order in the error terms. It may represent either vectors or matrices. The context of each expression will allow us to deduce what it refers to in each case. Along the developed methodology several assumptions have been considered, leading to different sources of error. The objective of this section is to address the order of the error associated to each approximation. The analysis will be carried out considering separately each assumption, deriving the corresponding order of magnitude of the error.

7.1. Approximation of $x^{(n-1)}(t)$ in $[t_k, t_{k+1}]$ by interpolation

In Eqs. (21), the derivation of terms $x_k^{(n)}$, $\dot{x}_k^{(n)}$ requires to solve the integrals with the transient solution of the $(n-1)$th iteration, $x^{(n-1)}(t)$. To this end, the function $x^{(n-1)}(t)$ is approximated by 3rd order cubic splines using displacements and velocities at both limits of the interval, i.e. $t = t_k$ and $t = t_{k+1}$. Under this assumption, Eq. (24) can be written in terms of the error order as

$$x^{(n-1)}(t) = N_1(\xi)x_k^{(n-1)} + D_1(\xi)\Delta t \dot{x}_k^{(n-1)} + N_2(\xi)x_{k+1}^{(n-1)} + D_2(\xi)\Delta t \dot{x}_{k+1}^{(n-1)} + \mathcal{O}(\Delta t^4) \quad (91)$$
Table 2: 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 $p = 20$. $\zeta$ stands for the damping ratio of the single dof oscillator of frequency $\omega = 2\pi/T$.

| Damping ratio | Truncation order | Stability boundaries |
|---------------|------------------|----------------------|
| $\zeta = 0.000$ | $m_a = 2$        | $0.0000 < \Delta t_0/T < 0.2579$ |
|               | $m_a = 4$        | $0.2644 < \Delta t_0/T < 0.5405$ |
|               | $m_a = 6$        | $0.0000 < \Delta t_0/T < 0.2579$ |
|               | $m_a = 8$        | $0.2749 < \Delta t_0/T < 0.7279$ |
| $\zeta = 0.005$ | $m_a = 2$        | $0.0000 < \Delta t_0/T < 0.2791$ |
|               | $m_a = 4$        | $0.0000 < \Delta t_0/T < 0.5406$ |
|               | $m_a = 6$        | $0.0000 < \Delta t_0/T < 0.3741$ |
|               | $m_a = 8$        | $0.0000 < \Delta t_0/T < 0.7279$ |
| $\zeta = 0.050$ | $m_a = 2$        | $0.0000 < \Delta t_0/T < 0.3023$ |
|               | $m_a = 4$        | $0.0000 < \Delta t_0/T < 0.5421$ |
|               | $m_a = 6$        | $0.0000 < \Delta t_0/T < 0.4766$ |
|               | $m_a = 8$        | $0.0000 < \Delta t_0/T < 0.7343$ |
| $\zeta = 0.500$ | $m_a = 2$        | $0.0000 < \Delta t_0/T < 0.3874$ |
|               | $m_a = 4$        | $0.0000 < \Delta t_0/T < 0.5312$ |
|               | $m_a = 6$        | $0.0000 < \Delta t_0/T < 0.6156$ |
|               | $m_a = 8$        | $0.0000 < \Delta t_0/T < 0.7407$ |

Plugging Eq. (91) into the integrals of Eq. (21) and following the same steps, we can rewrite Eq. (22) in simplified form in terms of the error as

$$X^{(n)}_{k+1} = X^{(n)}_k + \|M^{-1}C\| \mathcal{O}(\Delta t^p), \quad n \geq 1, \quad k \geq 0$$  \hspace{1cm} (92)

where $X^{(n)}_{k+1}$ denotes the exact solution of the $n$th iteration at instant $t_{k+1}$ and $X^{(n)}_{k} = T X^{(n)}_k + \alpha X^{(n-1)}_k + \beta X^{(n-1)}_{k+1}$ stands for the numerical approximate value. The exact solution is by definition

$$U(t_k) = \sum_{n=0}^\infty X^{(n)}_k$$  \hspace{1cm} (93)

Therefore, from Eqs. (92) and (93), it can be established that the order of the error due to the interpolation of $x^{(n-1)}(t)$ is proportional, something that can be written as

$$\|U(t_k) - U_k\| \leq \|M^{-1}C\| \mathcal{O}(\Delta t^p)$$  \hspace{1cm} (94)

for some norm $\|\cdot\|$. The above order of accuracy only measures the error of the approximation given by Eq. (91). Truncation of series involved in matrices $a(\Delta t)$ and $b_k(\Delta t)$ also give rise to errors, whose order of magnitude will be estimated in the following points.

7.2. Approximation of $\delta a(\Delta t_0)$ by truncation

As stated in Sec. 7.1, truncation of series is needed for computation of the incremental part, $\delta a(\Delta t_0) = a(\Delta t_0) - I_{2N}$ in Eq. (27). Let us denote by $\delta a(\Delta t_0)$ to the exact incremental matrix after retaining all terms of the series and by $\delta a(\Delta t_0)$ to the matrix numerically determined. Remind that $\Delta t_0 = \Delta t/2^p$ represents the reduced time step, with $p = 20$. Then, Eq. (76) can be written as

$$\delta a(\Delta t_0) = (I_{2N} - \beta)^{-1}(T + \alpha) - I_{2N} = \delta T + \alpha + \delta \beta + \delta \beta \delta T + \delta \beta \alpha$$  \hspace{1cm} (95)

where $\delta T$, $\alpha$, $\delta \beta$ stand for the exact standard of the matrices, which in turn can be expanded as function of the approximate matrices using the parameter $m_a$, yielding

$$\delta T = \delta T_a + \mathcal{O}(\Delta t_0^{m_a+2}) \mathcal{O}(\Delta t_0^{m_a+3}) \mathcal{O}(\Delta t_0^{m_a+4})$$

$$\delta \beta = \sum_{n=1}^\infty \beta^n \alpha_n + \mathcal{O}(\beta^n+1)$$  \hspace{1cm} (96)
The minimum order of the error in the three matrices is $O(\Delta t_{m_\alpha}^{m_\alpha+2})$. Hence, plugging Eq. (100) into Eq. (95) we have

$$
\delta a(\Delta t_0) = \delta a(\Delta t_0) + O(\Delta t_{0}^{m_\alpha+2}) + \|M^{-1} C\|^{r_a+1} O\left(\Delta t_{r_a+1}\right) = \delta a(\Delta t_0) + O\left(\Delta t_{m_\alpha+2}\right) + \|M^{-1} C\|^{r_a+1} O\left(\Delta t_{r_a+1}\right) \tag{97}
$$

Considering the common values $m_\alpha = 2$, $p = 20$ and $r_a = 2$, the error in the computation of $a(\Delta t)$ results to be of the order

$$
\|\delta a(\Delta t_0) - \delta a(\Delta t_0)\| \leq 8.27 \times 10^{-25} O\left(\Delta t^4\right) + 8.67 \times 10^{-19} \|M^{-1} C\|^3 O\left(\Delta t^3\right) \tag{98}
$$

which represents a very small error in terms of computer accuracy.

7.3. Approximation of $b_k(\Delta t)$ by truncation

We will now study the order of the error committed in the computation of the vector $b_k(\Delta t)$, associated to the nonhomogeneous solution. In the context of this section, we will denote by $b_k(\Delta t)$ the exact nonhomogeneous term and reserve the notation $b_k(\Delta t)$ for the approximate numerical result. In this vector, we find several sources of error:

(i) In Eq. (10) the vector $f(t)$ is approximated in $[t_k, t_{k+1}]$ by 3rd order Laplace polynomials.

(ii) In Eq. (54) The inverse matrix $(I_{2N} - \beta)^{-1}$ is approximated by a Neumann series up to the $r_b$-th order.

(iii) The truncation of both matrices $\beta$ and $L$ also introduces an error, controlled by the parameter $m_b$.

First, let us see the order of the error (i), as consequence of the approximation of the external forcing vector $f(t)$ shown in Eq. (10) by polynomial interpolation in the interval $[t_k, t_{k+1}]$. Since the type of interpolation functions are 3rd order Lagrange polynomials, Eq. (10) can now be written as

$$
f(t) = \sum_{i=1}^{4} L_i \left( \frac{t - t_k}{t_{k+1} - t_k} \right) f\left( t_k + \frac{(i - 1)\Delta t}{3} \right) + O(\Delta t^4), \quad t_k \leq t \leq t_{k+1} \tag{99}
$$

where $L_i(\xi)$, $1 \leq i \leq 4$ stand for the 3rd order Lagrange polynomials. The aim of this interpolation is the computation of the integrals within Eqs. (12), whose order of approximation results

$$
\left\{ \begin{array}{l} \int_{t_k}^{t_{k+1}} H(t_{k+1} - t) M^{-1} f(t) \, dt \\ \int_{t_k}^{t_{k+1}} G(t_{k+1} - t) M^{-1} f(t) \, dt \end{array} \right\} = L g_k + O(\Delta t^5) \tag{100}
$$

where $g_k$, defined in Eq. (10), contains the forcing terms at the data points $t_k + (i - 1)\Delta t/3$, $i = 1, 2, 3, 4$. Therefore the exact form of the vector $b_k(\Delta t)$ is

$$
b_k(\Delta t) = (I_{2N} - \beta)^{-1} \left[ L g_k + O(\Delta t^5) \right] \tag{101}
$$

Now, let us see how the truncation of the series in matrices $\beta$ and $L$ affect to the order of the error. Considering the parameters $r_b$ and $m_b$,

$$
(I_{2N} - \beta)^{-1} = \sum_{n=0}^{r_b} \beta^n + O\left(\|\beta\|^{r_b+1}\right), \quad \beta = \beta_b + O(\Delta t^{m_b+2}), \quad L = L_b + O(\Delta t^{m_b+3}) \tag{102}
$$

xx
8. Computation effort

In this section an estimation of the computational cost by counting the number of required operations will be addressed. In particular the number of matrix multiplications of the proposed method will be counted and compared with other explicit methods: the Modified Precise Integration Method (MPIM) the 4th order Runge–Kutta method (RK4). The computational cost of the calculation of $M^{-1}$ will not be considered, because it is a common requirement for the three methods.

8.1. Proposed method (PER)

The proposed method is based on the perturbation of the damping terms, hence it will be called the PER method in this section. We will distinguish between the two clearly differentiated stages: (1) a previous set-up of the system matrices, focused on the computation of main matrices $a$ and $b_k$ and (2) the computation of the solution at the different time instants by the iterative procedure, $U_1, \ldots, U_k$, where $k = t_{\text{max}} / \Delta t$. In order to derive a closed form of the operations count, let us establish some hypotheses

- The parameters $m_a$ and $m_b$, which measure the maximum order of $\Delta t$ in the computation of $a$ and $b_k$, are both even numbers and they verify that $m_b \geq m_a \geq 2$.
- The series $I_{2N} + \beta + \beta^2 + \cdots + \beta^r$ can be evaluated using $r/2$ matrix products by the algorithm

$$I_{2N} + \beta + \beta^2 + \cdots + \beta^r + \cdots = I_{2N} + \beta + \beta^2(I_{2N} + \beta + \beta^2(I_{2N} + \beta + \beta^2(I_{2N} + \cdots)))$$

consequently the parameters $r_a$ and $r_b$ will be even numbers.

| Source of error | Order of error, $\|\mathbf{M}(t_k) - \mathbf{U}_k\|$ | Description |
|-----------------|-----------------------------------------------|-------------|
| Sec. 6.3        | $\|M^{-1}C\| \mathcal{O}(\Delta t^5)$         | Approximation of $x(t) \text{ by cubic splines}$ |
| Sec. 7.2        | $\mathcal{O}(\Delta t^{m_b+2}) + \|M^{-1}C\| \mathcal{O}(\Delta t^{r_b+1})$ | Approximation of $a(\Delta t)$ |
| Sec. 7.3        | $\mathcal{O}(\Delta t^{m_a+2}) + \|M^{-1}C\| \mathcal{O}(\Delta t^{r_a+1}) + \mathcal{O}(\Delta t^2)$ | Approximation of vector $b_k(\Delta t)$ |

Table 3: Order of error of the proposed method from the different sources of approximation

where $\beta_k$ and $L_0$ are defined in Eq. (33). It has been shown in Eq. (51) that, if the ratio $\Delta t / T$ lies within certain range, $\beta$ can be considered proportional to the damping matrix and to the time step.

$$b_k(\Delta t) = (I_{2N} - \beta)^{-1} \left[ L_0 g_k + \mathcal{O}(\Delta t^5) \right]$$

$$= \sum_{n=0}^{r_a} \beta^n_0 + \mathcal{O}(\Delta t^{m_a+2}) + \|M^{-1}C\| \mathcal{O}(\Delta t^{r_a+1}) \left[ L_0 g_k + \mathcal{O}(\Delta t^5) \right]$$

$$= \sum_{n=0}^{r_b} \beta^n_0 \cdot L_0 g_k + \mathcal{O}(\Delta t^{m_b+2}) + \|M^{-1}C\| \mathcal{O}(\Delta t^{r_b+1}) + \mathcal{O}(\Delta t^5)$$

$$\equiv b_k(\Delta t) + \mathcal{O}(\Delta t^{m_b+2}) + \|M^{-1}C\| \mathcal{O}(\Delta t^{r_b+1}) + \mathcal{O}(\Delta t^5) \tag{103}$$

A summary of the order of accuracy obtained from each error source is presented in Table 3. For practical purposes, we will consider fixed the parameters $m_a = 2$ and $r_a = 2$ for the determination of $a(\Delta t)$. In view of the results in Table 3 the order of approximation would be theoretically $\mathcal{O}(\Delta t^5)$ for these particular values, although in practice these terms are affected by such a small number that the computation $a(\Delta t)$ with the precise method of Sec. 7.1 can be considered exact for computer precision. From the accuracy point of view, the other orders shown in Table 3 are more interesting since they show a dependence on the damping level, further affected by the parameter $r_b$, which will explain the behavior of the response error depicted later in the numerical simulations.
Some values that can be fixed are \( m_a = r_a = 2 \) and \( p = 20 \).

### 8.2. Modified Precise Integration Method (MPIM)

As shown in Sec. 5.1, a precise computation of the matrix \( \mathbf{a}(\Delta t) \) must be carried out using the \( 2^p \) algorithm. These feature is shared with the well-known Precise Integration Method \[17, 18, 21\], which make it a direct competitor from the computational point of view. The transient problem can be written under the state-space formalism

\[
\frac{d\mathbf{u}}{dt} = \mathbf{W}\mathbf{u}(t) + \mathbf{h}(t)
\]

where

\[
\mathbf{U}(t) = \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{u}(t) \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{O}_N & \mathbf{I}_N \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix}, \quad \mathbf{h}(t) = \begin{bmatrix} 0 \\ \mathbf{M}^{-1}\mathbf{f}(t) \end{bmatrix}
\]

Considering the time step \( \Delta t \), the iterative scheme yields

\[
\mathbf{U}_{k+1} = e^{\mathbf{W}\Delta t}\mathbf{U}_k + \int_{t_k}^{t_{k+1}} e^{\mathbf{W}(t_{k+1}-\tau)}\mathbf{h}(\tau)\,d\tau
\]

The main contribution of the Precise Integration Method is the proposal of a highly efficient computation of the exponential matrix. Denoting by \( \mathcal{H}(t) = e^{\mathbf{W}t} \), then

\[
\mathcal{H}(\Delta t) = [\mathcal{H}(\Delta t/2^p)]^{2^p}
\]
The Runge-Kutta method approximates the value of the function \( U(t_k + \Delta t) \) from the previous step \( U(t_k) \) in the ordinary differential equation \( \dot{U}(t) = F(U,t) \), where the vector field

\[
F(U,t) = WU + h(t)
\]

has already been introduced in Eq. \( \text{163} \) for oscillatory linear damped systems. In particular, the well-known fourth-order Runge-Kutta method (RK4, \( \text{13, 16} \)) achieves a truncation error at each time-step of

\[
C_{\text{MPIM}} = [18 + 8p(1 + g)] N^3 + [4g + 4t_{\text{max}}/\Delta t] N^2
\]
Table 6: Example 1. Natural frequencies $\omega_j$ and modal damping ratios $\xi_j = C'_{jj}/2\omega_j$ (relative to the damping parameter $\zeta$)

| Mode, $j$ | 1   | 2   | 3   | 4   | 5   | 6   |
|-----------|-----|-----|-----|-----|-----|-----|
| Natural frequency, $\omega_j$ (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  |
|-----------|-----|-----|-----|-----|-----|-----|
| $\omega_j$ (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 |

Table 6: Example 1. Natural frequencies $\omega_j$ and modal damping ratios $\xi_j = C'_{jj}/2\omega_j$ (relative to the damping parameter $\zeta$)

8.4. Comparison

In view of the computational cost analysis, the following conclusions can be drawn:

- As the size of the system increases, the terms which really matters are those proportional to $N^3$. Comparing the proposed method with the MPIM, the former will be numerically more efficient than the latter if

$$
\lim_{N \to \infty} \frac{C_{\text{PER}}}{C_{\text{MPIM}}} = \frac{33 + 4(r_a + r_b) + 8p + m_b}{18 + 8(1 + g)p} < 1
$$

(113)

For instance, considering $m_a = r_a = 2$ and $p = 20$, Eq. (113) leads to

$$
23 + m_b + 4r_b > 160g
$$

(114)

where $g$ stands for the number of Gauss integration points, which in general can be taken between 3 and 5 [21].

- When the number of iterations is small compared to the system size, the Runge-Kutta method is the most effective. It is very favorable the fact that no previous operations are needed to build the algorithm matrices (to construct the iterative scheme only costs $2N^3$ multiplications). However, it is known that the RK4 method has stability boundaries and requires relatively small time steps. This is detrimental to the computational efficiency since the computational cost per step is 4 times slower than the other two methods considered in this section (PER and MPIM). Thus, a high number of steps can tip the balance in favor of these latter methods.

9. Numerical examples

9.1. Example 1. Discrete system

We will first consider a numerical example consisting of a 12-dof discrete lumped spring–mass system shown in Fig. 3 (op). Mass, rigidity and damping coefficient are respectively $M = 1$ kg, $K = 100$ N/m and $C = 2\sqrt{KM\zeta}$, where $\zeta$ 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 6. As $\zeta$ increases, the coefficients of the damping matrix also increase linearly. Table 6 also lists the relationships between the modal damping ratios and the parameter $\zeta$, i.e. $\xi_j/\zeta$, where $\xi_j = C'_{jj}/2\omega_j$ and $C'_{jj}$ denotes the $j$th main diagonal entree of the modal damping matrix, $C' = \Phi^T C \Phi$. The minimum period of the structure is $T = 0.32$ s. Although the damping model is non-proportional, 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 3rd dof and it reproduces a multiharmonic input affected by a Gaussian weight. The mathematical model and the
Figure 3: 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.

corresponding graph are shown in Fig. 3(bottom) and it is formed by 4 harmonic functions with different amplitudes. The proposed method will be compared with some existing iterative numerical methods (listed below). The time step $\Delta t$ is a common parameter for all the methods.

**Runge–Kutta (RK4).** The fourth order Runge-Kutta method (RK4, [15, 16]), already presented in Sec. 8.3, produces a truncation error at each time-step of the order $O(\Delta t^5)$ and an overall error of order $O(\Delta t^4)$. The main computational cost is caused by the evaluation of the vector field $F(U, t) = W U + h(t)$ four times in each iteration.

**Newmark method (NEW).** The Newmark integration scheme [3, 54], also known as Newmark–$\beta$ method, is based on the application of the extended mean value theorem in the estimation of displacements and velocities, $u_{k+1}, \dot{u}_{k+1}$, from accelerations. For that, two parameters $\gamma$ and $\beta$ 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 (WIL).** The Wilson-$\theta$ method [54, 55] 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 [55]. Both Newmark’s and Wilson’s method are examples of implicit methods, which requires updating the system matrices in each iteration.

**Bathe method (BATHE).** The Bathe’s method [9–11] 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 $\gamma$ is a parameter. Two different approaches are applied within each sub-step: the trapezoidal rule is used for the first sub-step and the 3-point Euler backward method for the second one. For the current numerical examples, the case $\gamma = 1/2$ will be consider, which makes both time sub-steps of equal length, exhibiting in addition unconditional stability.

**Modified Precise Integration Method (MPIM).** [17, 18, 21] The Modified Precise Integration Method combines two techniques: on one hand, it uses the original approach of Zhong [17] for calculating the exponential matrix in the homogeneous solution based on the $2^p$ algorithm. On the other hand, the integral of the non-homogeneous term are accomplished using the Gauss quadrature method with $g$ integration points [21]. In reference [21], 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.
**Proposed method.** As described in the theoretical developments, several parameters can be tuned in the proposed method. Along the numerical examples the main matrix $a(\Delta t)$ 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. Parameters $m_b$ and $r_b$ need to be fixed in order to obtain the non-homogeneous term $b_k(\Delta t)$. As will be seen below, the appropriate values of these parameters depend on the level of damping and the choice of time step. Therefore, they will be left free for the time being.

![Figure 4: (Example 1) Time-domain numerical solutions for the different methods at dof $u_1(t)$. Dimensionless damping parameter $\zeta = 0.10$. Time-step $\Delta t = 0.24 s = 0.75T$. Newmark-$\beta$ method with parameters $\gamma = 1/2$, $\beta = 1/4$. Wilson-$\theta$ method is calculated for $\theta = 1.4$. Bathe’s method is implemented with $\gamma = 1/2$. Modified PIM (MPIM) uses $g = 4$ Gauss quadrature points in integration of nonhomogeneous term. Order of truncation in proposed method in the non-homogeneous term: $m_b = 8$ and $r_b = 4$. The reference solution has been evaluated using RK4 with a time-step $\Delta t/500$. (a) Displacement $u_1(t)$, (b) Local relative errors of displacements. (c) Velocity of 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.](image-url)
In fig. 4 the transient problem for the current example has been solved using the six different approaches presented above. Figs. 4(a) and 4(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 $0 \leq t \leq t_{\text{max}} = 40$ s, with a time-step of $\Delta t = 0.24 = 0.75T$ s. Damping parameter has been taken $\zeta = 0.10$, something that leads to a set of modal damping ratios in the range $0.97 \leq \xi < 18.69\%$. The relationship between the spectral radius of matrices $M^{-1}C$ and $\sqrt{M^{-1}K}$, $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) = 0.815$ 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. 4(b) and 4(d). Since the time step chosen 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 $\Delta t$ used in fig. 4 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 $g$ has been obtained using $\Delta t/24 = 0.8617 < 1$, guaranteeing the convergence although with a value very close to 1. We take $r_{\text{g}} = 4$ 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 revealed also that for $g > 4$ the accuracy barely change. 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. 4(e) and 4(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. 5 where the displacement and velocity response together with the corresponding relative errors have been plotted for a 10–times smaller time step respect to that used for fig. 4 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. In the results of fig. 5 the values $m_b = 4$ and $r_{\text{g}} = 4$ are chosen. In general, the relative errors drop by 2 to 4 orders of magnitude with respect to those shown in fig. 4. Moreover, the RK4 method is 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 one that, among all the simulated approaches, yields the highest accurate results. The error in both displacement and velocity improve appreciably in all methodologies within the range $30 < t < 40$ s, where the non-homogeneous terms are reduced to practically zero.

We will now focus on studying the influence of different parameters in the global error. The proposed method depends mainly on three relevant parameters,

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

(ii) The damping level, quantified throughout the entries of the matrix $C$, becomes of special importance in the proposed method since this latter is constructed on the basis of its perturbation. 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 $\zeta$. In the results the dimensionless ratio $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K})$ will be used as a relative measure of the damping level.

(iii) The truncation orders, $m_b$ and $r_{\text{g}}$, introduced to construct the non–homogeneous term $b_b(\Delta t)$ in Eqs. (22) and (31).
Let us define a measure of the total error (global error) over the entire time range. Given an interval 
\([0, t_{\text{max}}]\), a time step \(\Delta t\) and a function \(y(t)\) defined at all points \(y(t_k), 0 \leq k \leq k_{\text{max}}\), \(t_k = k\Delta t\), with 
\(k_{\text{max}} = t_{\text{max}}/\Delta t\). Then we define the following function-norm \(\|\cdot\|\) as

\[
\|y(t)\| = \sqrt{\sum_{k=0}^{k_{\text{max}}} y^2(t_k)} \tag{115}
\]

and the global error of the \(j\)th dof displacement and velocity can then be defined as

\[
e(u_j, \Delta t) = \frac{\|u_j(t) - u_{j,\text{ref}}(t)\|}{\|u_{j,\text{ref}}(t)\|}, \quad e(\dot{u}_j, \Delta t) = \frac{\|\dot{u}_j(t) - \dot{u}_{j,\text{ref}}(t)\|}{\|\dot{u}_{j,\text{ref}}(t)\|} \tag{116}
\]

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. (a) and (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 \(\rho (M^{-1} C)/(\sqrt{M^{-1} K}) = 0.10\), corresponding to \(\zeta = 0.0123\). The left-side plots (low damping) can be compared with those of the right side (high damping), figs. (c) and (d), which show the same variables but for a higher level of damping, i.e. \(\rho (M^{-1} C)/(\sqrt{M^{-1} K}) = 1.00\), calculated from \(\zeta = 0.123\). In view of the results, the proposed method shows evidence of high accuracy compared to the other methods studied. Although it seems to reach an error of the order of \(10^{-7}\) below which it fails to decrease, unlike the rest of the methods which do undergo a monotonic decreasing as \(\Delta t/T\) approaches to zero. 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. 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 figs. 6 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 \[49\]. As shown in Sec. 4, the spectral radius $\rho(\beta)$ is strongly dependent on the time-step and on the damping level. In the algorithms developed in Sec. 5 for the evaluation of the scheme matrices, the $\beta$-matrix needs to be computed twice: the first one, (matrix $\beta_{a\mu}$, for the computation of the matrix $a(\Delta t)$ and the second one (matrix $\beta_b$) for the evaluation of the non-homogeneous terms within $b_k(\Delta t)$. It is indeed this latter which somewhat limits the range of validity and may diverge for relatively high values of $\Delta t/T$ as seen in the example. 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.

**LIGHT DAMPING**

|   | $u(t)$-Error, $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) = 0.100$ | $\dot{u}(t)$-Error, $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) = 0.100$ |
|---|---|---|
| (a) | ![Light damping](image1) | ![Light damping](image2) |
| (b) | ![Light damping](image3) | ![Light damping](image4) |

**HIGH DAMPING**

|   | $u(t)$-Error, $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) = 1.000$ | $\dot{u}(t)$-Error, $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) = 1.000$ |
|---|---|---|
| (c) | ![High damping](image5) | ![High damping](image6) |
| (d) | ![High damping](image7) | ![High damping](image8) |

Figure 6: Relationship between global error of dof $\# 2$, $e(\dot{u}_2, \Delta t)\, e(\dot{u}_2, \Delta t)$ and time step $\Delta t/T$ (example 1), for two cases of damping level: (a) and (b) global error plots for $\zeta = 0.012$, $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) = 0.10$. (c) and (d) global error plots for $\zeta = 0.123$, $\rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) = 1.00$. Newmark–$\beta$ method with parameters $\gamma = 1/2$, $\beta = 1/4$. Wilson–$\theta$ method is calculated for $\theta = 1.4$. Bathe’s method is implemented with $\gamma = 1/2$. Orders of truncation for the proposed method in the non-homogeneous term: $m_b \in \{4, 24\}$ and $r_b \in \{2, 12\}$.

As expected, the error increases with the time step also for the proposed method. However, different configurations between the $r_b$ and $m_b$ parameters result in different error distributions. In order to identify
the influence of these parameters, different simulations have been carried out for \( r_b = \{2, 12\} \) and \( m_b = \{4, 24\} \). The observed pattern in the error evolution can be partly explained with the theoretically derived order of accuracy in Table 3, where the error order associated to the non-homogeneous terms

\[
O(\Delta t^{m_b+2}) + \|M^{-1}C\|^{r+1}O(\Delta t^{r+1})
\]  

(E1)

(E2)

For lightly damped systems, figs 6(a) and 6(b), it is verified that \( \rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) \ll 1 \). Thus, the term (E2) in Eq. (117) is very small compared to term (E1) and the error is actually governed this latter, i.e. \( O(\Delta t^{m_b+2}) \). This explains why the benefit of taking a high value such as \( m_b = 24 \), respect to \( m_b = 4 \), is only appreciated for high values of \( \Delta t/T \). Moreover, figs. 6(a) and 6(b) are hardly insensitive to the parameter \( r_b \). However, the result of increasing the damping is a proportional increase in the spectral radius \( \rho(\beta) \). If the coefficient \( r_b \) is not changed, then the computation of \( (I - \beta\beta)^{-1} \) by Neumann series is poorer and the error will increase. This fact can be seen in figs. 6(c) and 6(d) where the error for the \( r_b = 2 \) curves increases significantly with respect to those of the lightly damped case. If we seek to avoid this loss of accuracy, the number of \( r_b \) terms in the Neumann series must be increased. Thus, increasing \( r_b \) from \( r_b = 2 \) to \( r_b = 12 \), leads to a substantial decreasing in the error. Furthermore, the slope of the curves (close related to the order of precision) is also slightly affected by increasing \( r_b \). This validates the theoretical result shown in Eq. (117), which predicts a precision order dependent on the parameter \( r_b \).

The underlying nature of the proposed method, based on the perturbation of the dissipative model, suggests that the results will depend strongly on damping. The trend observed in figs. 7 makes us wonder how the quality of our proposal will evolve as we increase the damping level. To this end, fig. 7 shows the relative global error for two time steps, \( \Delta t/T = 0.05, 0.50 \), respectively. In all performed simulations, the proposed method exhibits the lowest error and consequently the most accurate results for lightly damped systems. It is also observed that the choice of \( r_b \) does not affect the error when the damping is small, something that, as above, can be explained by the fact that the error term (E2) in Eq. (117) is negligible with respect to (E1). In Figs 7(a) and 7(c) the spectral radius \( \rho(\beta) \) has also been plotted, showing its 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 \( \rho(\beta) = 1 \) it is useless to increase \( r_b \) since the Neumann series is divergent. It can be seen in the four graphs of fig. 7 the divergence of the method when this threshold is exceeded. The other methods under study are also insensitive to damping at low values of \( \rho(M^{-1}C) \). 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 \( \rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) \approx 8 \). On the right side, 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. We have purposely left the same scale on the ordinate (error) axes for all graphs. This allows to observe at a simple glance the effect of multiplying the time step by 10, from \( \Delta t/T = 0.05 \) in figs. 7(a) and 7(b) to \( \Delta t/T = 0.50 \) in figs. 7(c) and 7(d). As expected, all methods exhibit an increasing error, although the proposed method for these particular values still yields the best numerical results for light and moderate damping problems.

9.2. Example 2. Continuous system

In this example, the application of the proposed numerical method to continuous dynamic problems is explored. A cantilever beam of fig. 8 supported by two elastic constraints with viscous dampers is considered. The total length is \( l = 3 \) m and the cross-section is steel-made with bending stiffness \( EI = 437.5 \) kNm\(^2\) and total mass \( m_0 = 235.5 \) kg. A finite element model based on Euler-Bernoulli beam theory with \( n_e \) elements is used. The stiffnesses of the supports are \( K_A = 20EI/l^4 = 324 \) kN/m and \( K_B = 10EI/l^3 = 162 \) kN/m

xxx
\( \Delta t/T = 0.05 \)

\( \frac{\Delta t}{T} = 0.05 \)

\( \frac{u(t)}{\Delta t} \text{ Error.} \Delta t/T = 0.05 \)

\( \frac{\Delta t}{T} = 0.50 \)

\( \frac{u(t)}{\Delta t} \text{ Error.} \Delta t/T = 0.50 \)

Figure 7: Relationship between global error of dof \( \# 2, e(u_2, \Delta t), e(\dot{u}_2, \Delta t) \) and damping level, measured as \( \rho(M^{-1}C)/\rho(\sqrt{M^{-1}K}) \) (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 \( \Delta t/T = 0.5 \). Newmark–\( \beta \) method with parameters \( \gamma = 1/2, \beta = 1/4 \). Wilson–\( \theta \) method is calculated for \( \theta = 1.4 \). Bathe’s method is implemented with \( \gamma = 1/2 \). Orders of truncation for the proposed method in the non-homogeneous 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 \( \beta_b \).

The damping coefficients are respectively \( c_A = 2m_0\omega_r\zeta_A \) and \( c_B = 2m_0\omega_r\zeta_B \), where \( \omega_r = \sqrt{EI/m_0l^2} = 8.29 \text{ rad/s} \) is a reference frequency. The damping parameters \( \zeta_A, \zeta_B \) will be fixed later for each simulation. A point force \( f(t) \) is applied downwards at the free right-end of the beam, as shown in fig. S(left). The force vs time distribution is shown in the graph on in fig. S(right), where

\[
f(t) = \begin{cases} 0 & t < t_c \\ f_0 & t \geq t_c \end{cases}
\]

The following parameters are taken: \( t_c = 0.01 \text{ s} \) and \( f_0 = 1 \text{ kN} \). In fig. S some simulations of the beam have been carried out using \( n_e = 24 \) elements and \( N = 48 \) degrees of freedom. The structure has a minimum period of \( T = 3.1013 \times 10^{-5} \text{ s} \). The damping parameter for both elastic supports has been taken as \( \zeta_A = \zeta_B = 0.50 \) resulting modal damping ratios in the range \( 0.6% \leq \xi_j \leq 11.2% \) in the six first modes, while higher modes present values lower than 0.4%. The time domain response has been computed for both displacement and velocity \( u(t) \) and \( \dot{u}(t) \) at the right-end of the beam, downwards, as indicated in fig. xxxi.
Figure 8: Example 2. Continuous system

- Figs. 9(a) and 9(b) represent the reference solution (displacement and velocity, respectively) calculated using a time step of $\Delta t/500$. Two different cases corresponding to two time steps have been considered, namely

**Case $\Delta t/T = 0.4$ [figs. 9(c), 9(d), 9(e) and 9(f)]:** The truncation orders in the non-homogeneous term of the proposed method are chosen as $r_b = 2$ and $m_b = 4$ and the spectral radius of the $\beta_b$ matrix is $\rho(\beta_b) = 0.0011$. In general, something common to all considered methods is the fact that errors in displacements are significantly smaller than those of the velocities. It is observed that the methods based on the finite difference approximation are those ones with the highest errors. The displacement error curves show similar accuracy in the calculation of $u(t)$ for the MPIM (using $g = 5$ integration points) and the proposed method, see Fig. 9(e). However, velocities calculated with the proposed method is reported to be considerably more accurate than the others, as shown in Fig. 9(f).

**Case $\Delta t/T = 1.5$ [figs. 9(g), 9(h), 9(i) and 9(j)]:** Increasing the time step obviously leads to increase the relative error in all methods. In particular, the RK4 method becomes unstable so that its results are not displayed in the plots. Again, results observed in displacements are more accurate than those of velocities. The precision of MPIM and that of the proposed method are very similar in the calculation of displacement, but the evaluation of velocities turns out to be more accurate with the proposed method. To achieve this accuracy, it has been necessary to increase the parameter $m_b$ up to the value $m_b = 36$. In general, increasing $m_b$ is the most effective when faced with a significant increase in time step. While, on the other hand, including more terms in the Neumann series (increasing $r_b$) is only efficient for cases where $\rho(\beta_b)$ is moderately high (but not reaching one). In case of study, since the spectral radius is $\rho(\beta_b) = 0.0033 \ll 1$, it is not necessary to increase $r_b$, and the value $r_b = 2$ results sufficient.
Figure 9: Numerical simulations of the continuous beam of Example 2 with $n_e = 24$ elements and $N = 48$ dof with a minimum period of $T = 3.1013 \times 10^{-5}$ s. The damping parameter of supports is $\zeta_A = \zeta_B = 0.50$. (a) and (b): Reference response, displacement $u(t)$ and velocity $\dot{u}(t)$ at the right-edge of the beam respectively. (c) and (d): Details (I) and (II) of the displacement and velocity for $\Delta t = 0.4T$. (e) and (f): relative error of displacement and velocity respectively, for $\Delta t = 0.4T$. (g) and (h): Details (I) and (II) of the displacement and velocity for $\Delta t = 1.5T$. (i) and (j): relative error of displacement and velocity respectively, for $\Delta t = 1.5T$. 

xxxiii
After several numerical examples simulated, the truncation order \( m_b \) of the \( \beta_b \) series contributes to a significant reduction of the error for relatively high values of the time step. However, for low or moderate values, no visible effect is perceived. In order to better understand the influence that the parameter \( m_b \) has on the solution, some numerical simulations have been performed by changing this parameter for different time steps. The results in Fig. 11 show that the errors of displacements seem to be practically insensitive to the parameter \( m_b \) for \( \Delta t/T < 1 \). Only from this value (in Fig. 11(a), the curves \( \Delta t/T = 1 \) and \( \Delta t/T = 2 \)), it is really effective to increase \( m_b \) to bound the error, although this latter stabilizes from a limit value of \( m_b \).

Thus, for instance, when \( \Delta t/T = 1 \) the error does not decrease for \( m_b \geq 16 \), while in the case \( \Delta t/T = 2 \), this limit is at \( m_b \geq 32 \). The velocity errors follows the same pattern as the displacements. Although the parameter \( m_b \) starts to be sensitive already for values for \( \Delta t/T = 0.5 \) onwards. Thus, for example, the error in the velocities for \( \Delta t/T = 2 \) stabilizes for \( m_b \geq 36 \). As can be seen in Fig. 11 although the error stabilizes, the stabilization value grows considerably with increasing time step, especially in the computation of the velocities.

The behavior observed in Fig. 11 makes useless and unnecessary to take very high values of \( m_b \). This is advantageous since instabilities have been identified in the evaluation of the series for very high values of \( m_b \), in the current example for the order \( m_b > 60 \). Although they are under study, it seems likely that they are due to the numerical computation of terms of the type \( A_j/(2j + 4)! \) with \( j \) very high.

Next, the theoretical results obtained about the computational effort of the proposed method will be validated. First, a comparison of the computational time of the main matrices of the iterative scheme in both the MPIM and the proposed method will be carried out. Let us denote by \( T_{\text{PER}}(N) \) to the computation time required to obtain both matrices \( a(\Delta t) \) and \( b(\Delta t) \) using \( N \) degrees of freedom and by \( T_{\text{MPIM}}(N) \) the same for the computation of \( H(\Delta t) \) and \( w_k(\Delta t) \), Eqs. (108) and (110), respectively. In the theoretical developments of Sec. 8 closed form expressions for the number of multiplications needed to obtain these matrices, depending on the number of degrees of freedom, have been determined for both methods. The corresponding computation time arises then from multiplying the counting expression by the unitary time per operation, \( T_u \) (measured in seconds). Considering only terms proportional to \( N^3 \), it yields

\[
\begin{align*}
T_{\text{PER}} & = T_u C_{\text{PER}} \approx T_u (33 + 4(r_a + r_b) + 8p + m_b) N^3 \\
T_{\text{MPIM}} & = T_u C_{\text{MPIM}} \approx T_u (18 + 8g) N^3
\end{align*}
\]

(119)

The proposed model and the MPIM have been run varying the number of dof \( N \), keeping fixed the next xxxiv
parameters: for the proposed method $m_a = 2$, $r_a = 2$, $p = 20$, $r_b = 2$, $\zeta_A = \zeta_B = 0.50$. The computation times have been monitored and plotted in fig. 11 as numerical simulations. From these numerical experiments, the parameter $T_u$ has been obtained by curve-fitting resulting an average of $T_u = 2.216 \times 10^{-9}$ seconds per operation. As it can be observed in fig. 11(a), the model of Eq. (119) results suitable to describe the computation time needed for the preparation of the main matrices in both methods, for the particular values of $m_b = 4$ and $g = 4$. In order to check the goodness of these expressions for other cases, we have plotted in fig. 11(b) the relation between computation times $T_{PER}/T_{MPIM}$, which for a high number of degrees of freedom should tend to

$$
\lim_{N \to \infty} \frac{T_{PER}}{T_{MPIM}} = \frac{33 + 4(r_a + r_b) + 8p + m_b}{18 + 8(1 + g)p}
$$

(120)

The graphical results shown in fig. 11(b) show that indeed the relationship between the computation times obtained from the numerical experiments fit the model of Eq. (120) quite accurately, reflecting that within the values considered in practice as adequate, the preparation of the matrices is more effective with the proposed method, which in general takes less than half the time long.

It is also interesting to compare the computation time considering the whole process of the algorithm, including the calculation of the model matrices and the iterative process that includes $k_{\text{max}} = t_{\text{max}}/\Delta t$ steps. As derived from the theory, the RK4 procedure relatively consumes hardly any computational time in the model preparation, although within the iterative stage, it takes 4 times longer in each step than the MPIM and the proposed methods. This difference is expected to be noticed in simulations with a large number of iterations in relation to the system size. In fig. 12 the total computation time for the six methods considered has been plotted as a function of the number of degrees of freedom, $N$. In addition, 4 different simulations with $k_{\text{max}} = \{10^2, 10^3, 10^4, 10^5\}$ iterations have been considered. With only $k_{\text{max}} = 10^2$ iterations, the computation of the main matrices of the algorithm takes most of the computation time, making both the MPIM and the proposed method the least efficient. Roughly, for $k_{\text{max}} > 10^3$ it can be seen that the RK4 method is slower than both the proposed method and the MPIM, something that is especially noticeable in fig. 12(d) with a large number of iterations. In this latter plot, it can be observed that the time used for the preparation of the algorithm matrices in the MPIM and the proposed method is no longer relevant with respect to that of iterative process. It is observed that in this two methods, the computation time is similar for systems with a large number of degrees of freedom.
10. Conclusions

This paper considers the numerical resolution of transient problems in 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, order of accuracy and computational effort 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 two numerical examples. The first one consists of a lumped-mass discrete 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. In the second example, the transient response in a continuous beam problem has been determined. The goodness of the proposed method has been validated with respect to the other methods and times of computation have been compared, verifying that the proposed method has important computational advantages.
11. Acknowledgments

This work was supported by the Ministerio de Ciencia e Innovación (Spain), grant number PID2020-112759GB-I00.

References

[1] M. A. Dokainish, K. Subbaraj, A survey of direct time-integration methods in computational structural dynamics-i. explicit methods, Computers & Structures 32 (1989) 1371–1386. doi:10.1016/0045-7949(89)90314-3

[2] K. Subbaraj, M. A. Dokainish, A survey of direct time-integration methods in computational structural dynamics-ii. implicit methods, Computers & Structures 32 (1989) 1387–1401. doi:10.1016/0045-7949(89)90315-5

[3] N. M. Newmark, A method of computation for structural dynamics, Journal of the Engineering Mechanics Division 85(EM3) (1959) 67–94.

[4] E. L. Wilson, A computer program for the dynamic stress analysis of underground structures, Tech. rep., Division of Structural Engineering Structural Mechanics, University of California, Berkeley. Report No.68-1 (1968).

[5] H. Hilber, T. Hughes, R. Taylor, Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics, Earthquake Engineering and Structural Dynamics 5 (3) (1977) 283–292.

[6] W. L. Wood, M. Bossak, O. C. Zienkiewicz, An alpha modification of newmark’s method, International Journal for Numerical Methods in Engineering 15 (10) (1980) 1562–1566.

[7] J. Chung, G. M. Hulbert, A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized-a Method, Journal of Applied Mechanics 60 (2) (1993) 371–375.

[8] S. Krenk, Energy conservation in newmark based time integration algorithms, Computer Methods in Applied Mechanics and Engineering 195 (44) (2006) 6110–6124.

[9] K.-J. Bathe, M. Baig, On a composite implicit time integration procedure for nonlinear dynamics, Computers and Structures 83 (31-32) (2005) 2513–2524.

[10] K.-J. Bathe, Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration scheme, Computers & Structures 212 (2019) 299–310.

[11] G. Noh, K.-J. Bathe, The Bathe time integration method with controllable spectral radius: The $\rho_\infty$-Bathe method, Computers & Structures 212 (2019) 289–298. doi:10.1016/j.compstruc.2018.10.008

[12] M. M. Malakhiyeh, S. Shojaei, K.-J. Bathe, The Bathe time integration method revisited for prescribing desired numerical dissipation, Computers & Structures 212 (2019) 289–298. doi:10.1016/j.compstruc.2018.10.008

[13] J. Butcher, Numerical Methods for Ordinary Differential Equations, 2nd Edition, John Wiley and Sons, 2008.

[14] R. L. Burden, J. D. Faires, Numerical Analysis, Brooks-Cole. Cengage Learning, 2001.

[15] W. X. Zhong, F. W. Williams, A precise time step integration method, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 208 (6) (1994) 427–430.

[16] Z. Wan-Xie, On precise integration method, Journal of Computational and Applied Mathematics 163 (1) (2004) 59–78, proceedings of the International Symposium on Computational Mathematics and Applications.

[17] T. C. Fung, Computation of the matrix exponential and its derivatives by scaling and squaring, International Journal for Numerical Methods in Engineering 59 (10) (2004) 1273–1286.

[18] J. Lin, W. Shen, F. Williams, A high precision direct integration scheme for structures subjected to transient dynamic loading, Computers & Structures 56 (1) (1995) 113–120.

[19] T. C. Fung, Z. L. Chen, Krylov precise time-step integration method, International Journal for Numerical Methods in Engineering 68 (11) (2006) 1115–1136.

[20] X. Q. ZHU, S. S. LAW, Precise time-step integration for the dynamic response of a continuous beam under moving loads 240 (2001) 962–970. doi:10.1006/jenm.2000.3184

[21] C. C. Caprani, A modal precise integration method for the calculation of footbridge vibration response, Computers & Structures 128 (2013) 116–127.

[22] A. H. Nayfeh, Perturbation Methods, Wiley-VCH, 2004.

xxvii
L. Meirovitch, G. Ryland II, Response of slightly damped gyroscopic systems, Journal of Sound and Vibration 67 (1) (1979) 1–19, cited By 39.

L. Meirovitch, G. Ryland, A perturbation technique for gyroscopic systems with small internal and external damping, Journal of Sound and Vibration 100 (3) (1985) 395–408, cited By 21.

K. Chung, C. Lee, Dynamic reanalysis of weakly non-proportionally damped systems, Journal of Sound and Vibration 111 (1) (1986) 37 – 50.

S. Peres-Da-Silva, D. Cronin, T. Randolph, Computation of eigenvalues and eigenvectors of nonclassically damped systems, Computers and Structures 57 (5) (1995) 883–891.

P. D. Cha, Approximate eigensolutions for arbitrarily damped nearly proportional systems, Journal of Sound and Vibration 288 (4-5) (2005) 813 – 827.

M. Lázaro, Eigensolutions of non-proportionally damped systems based on continuous damping sensitivity, Journal of Sound and Vibration 363 (C) (2016) 532–544.

E. M. Daya, M. Potier-Ferry, A numerical method for nonlinear eigenvalue problems application to vibrations of viscoelastic structures, Computers & Structures 79 (5) (2001) 533 – 541.

L. Duigou, E. M. Daya, M. Potier-Ferry, Iterative algorithms for non-linear eigenvalue problems. application to vibrations of viscoelastic shells, Computer Methods in Applied Mechanics and Engineering 192 (11-12) (2003) 1321 – 1335.

M. Lázaro, J. L. Pérez-Aparicio, Dynamic analysis of frame structures with free viscoelastic layers: New closed-form solutions of eigenvalues and a viscous approach, Engineering Structures 54 (2013) 69 – 81.

L. Zoghaib, P.-O. Mattei, Time and frequency response of structures with frequency dependent, non-proportional linear damping, Journal of Sound and Vibration 333c (3) (2014) 887 – 900.

M. Lázaro, Nonviscous modes of nonproportionally damped viscoelastic systems, Journal of Applied Mechanics (Transactions of ASME) 82 (12) (2015) Art. 121011 (9 pp).

M. Lázaro, C. F. Casanova, I. Ferrer, P. Martín, Analysis of nonviscous oscillators based on the damping model perturbation, Shock and Vibration 2016 (2016) Art. 368129 (20p).

M. Lázaro, Closed-form eigensolutions of nonviscously, nonproportionally damped systems based on continuous damping sensitivity, Journal of Sound and Vibration 413 (2018) 368–382.

R. H. Gallagher, Perturbation procedures in nonlinear finite element structural analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 1975, pp. 75–89.

B. Cochelin, N. Damil, M. Potier-Ferry, Asymptotic-numerical methods and pade approximants for non-linear elastic structures, International Journal for Numerical Methods in Engineering 37 (7) (1994) 1187–1213.

S.-L. Mei, C.-J. Du, S.-W. Zhang, Asymptotic numerical method for multi-degree-of-freedom nonlinear dynamic systems, Chaos, Solitons & Fractals 35 (3) (2008) 536–542.

M. Fafard, K. Henchi, G. Gendron, Application of an asymptotic method to transient dynamic problems, Journal of Sound and Vibration 208 (1) (1997) 73–99.

N. Berrahma-Chekroun, M. Fafard, J. J. Gervais, Resolution of the transient dynamic problem with arbitrary loading using the asymptotic method, Journal of Sound and Vibration 243 (3) (2001) 475–501.

W. Mansur, F. Loureiro, D. Soares, C. Dors, Explicit time-domain approaches based on numerical green functions computed by finite differences - the exga family, Journal of Computational Physics 227 (1) (2007) 851 – 870. doi:https://doi.org/10.1016/j.jcp.2007.08.028

J. H. Wilkinson, Algebraic Eigenvalue Problem, Oxford University Press, 1988.

T. Caughey, M. O’Kelly, Classical Normal Modes in Damped Linear Dynamic Systems, Journal of Applied Mechanics 32 (3) (1965) 583–588.

C. Moler, C. V. Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Review 45 (1) (2003) 3–49.

C. Huang, M. Fu, A modified precise integration method for long-time duration dynamic analysis. article id 9646017, Mathematical Problems in Engineering.

A. Leung, Steady state response of undamped systems to excitations expressed as polynomials in time, Journal of Sound and Vibration 106 (1) (1986) 145–151. doi:https://doi.org/10.1016/S0022-460X(86)80178-X URL:https://www.sciencedirect.com/science/article/pii/S0022460X8680178X

K.-J. Bathe, Finite Element Procedures, 2014.

E. Wilson, I. Farhoomand, K. Bathe, Nonlinear dynamic analysis of complex structures., Earthquake Engineering and Structural Dynamics 1 (1973) 241–252.