

A generalized algorithm framework for non-linear structural dynamics

George Papazafeiropoulos¹  · Vagelis Plevris² ·
Manolis Papadrakakis¹

Received: 2 March 2016 / Accepted: 21 July 2016
© Springer Science+Business Media Dordrecht 2016

Abstract The performance of a family of nonlinear generalized single step-single solve (GSSSS) time integration schemes is assessed by comparison of their results in terms of total energy and the agreement with respective results published in the literature. The nonlinear algorithms have been developed by their linear counterparts using a Newton–Raphson iterative procedure to ensure dynamic equilibrium inside each time step. A literature review of the available time integration schemes used for nonlinear problems and the family of linear GSSSS algorithms are presented along with several commonly used time integration algorithms as special cases. Afterwards, the nonlinear schemes are formulated, and outlined in an explicit flowchart, which describes the nonlinear integration procedure in detail. The nonlinear family of algorithms is applied to six benchmark problems involving the dynamic response of SDOF systems with various stiffness and damping properties, as well as to a 3dof structure representing finite element systems containing rigid connections, penalty factors and other such types of constraints. It is shown that the schemes with Continuous Acceleration formulation (such as the HHT-a method) perform in general better than the others, even with a large time step, which leads to reduced computational effort for the estimation of the nonlinear dynamic response with relatively little loss of accuracy.

✉ George Papazafeiropoulos
gpapazafeiropoulos@yahoo.gr

Vagelis Plevris
vagelis.plevris@hioa.no

Manolis Papadrakakis
mpapadra@central.ntua.gr

¹ Institute of Structural Analysis and Antiseismic Research, National Technical University of Athens, Zografou Campus, 15780 Athens, Greece

² Department of Civil Engineering and Energy Technology, Oslo and Akershus University College of Applied Sciences, Pilestredet 35, 0166 Oslo, Norway

Keywords Time integration scheme · Newton–Raphson · Single step · Single solve · Energy conservation

1 Introduction

The dynamic analysis of engineering structures under dynamic loading (earthquake, impact, etc.) with the finite element method results in a set of nonlinear differential equations as follows:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{p}(\mathbf{u}, \dot{\mathbf{u}}) = \mathbf{f}(t) \quad (1)$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{p}$ is the internal force vector, which is in general a nonlinear function of the displacements $\mathbf{u}$ and velocities $\dot{\mathbf{u}}$ and is equal to the sum of the forces in the structure due to stiffness and damping, and $\mathbf{f}$ is the external force vector which is a function of time t . In the case of a linear elastic structure with viscous damping, $\mathbf{p}(\mathbf{u}, \dot{\mathbf{u}})$ is equal to:

$$\mathbf{p}(\mathbf{u}, \dot{\mathbf{u}}) = \bar{\mathbf{K}}\mathbf{u} + \bar{\mathbf{C}}\dot{\mathbf{u}} \quad (2)$$

where $\bar{\mathbf{K}}$ is the stiffness matrix and $\bar{\mathbf{C}}$ is the damping matrix, both of them independent of the displacement and velocity. Linear equations of dynamic equilibrium of the form of (1) in which $\mathbf{p}(\mathbf{u}, \dot{\mathbf{u}})$ is given by (2) can be solved using various superposition methods in the time or frequency domain, which greatly simplify the problem. However, in dynamic analysis of nonlinear response, superposition cannot be used and one has to resort to direct time integration (or time stepping, or step-by-step) methods. In these methods the equilibrium equations are satisfied at discrete time points (or steps) of the loading and the response history. The response during each step is calculated from the displacement and velocity at the beginning of the step and from the history of loading during the step. Thus the response for each step is an independent analysis problem.

The most common characteristics of integration schemes concern stability, convergence, accuracy, requirement of a starting procedure, and overshooting. A time integration scheme should be unconditionally stable, at least second-order accurate in time, self-starting, and should have minimal numerical dissipation and dispersion and no more than zero-order displacement and velocity overshooting behavior.

Regarding linear dynamic response, accuracy is the main concern, since many time integration algorithms are unconditionally stable. However, algorithms which are unconditionally stable for linear dynamics, often lose this stability for nonlinear dynamics, and therefore numerical stability is of primary interest in nonlinear dynamic systems.

The simplest direct time integration method for dynamic analysis is the piecewise exact method in which the equation of motion is solved exactly for linear loading during each time step, in which it is assumed that the actual loading history has constant slope (Clough and Penzien 2003). Although the equation of motion is solved rigorously during each time step, the linear interpolation of the excitation function introduces some error into the calculated response; this can be eliminated either by reducing the length of the time step, or adjusting it so that the introduced loading history fits best the actual one.

The numerical direct time integration schemes can be classified as either explicit or implicit. An explicit scheme is one in which the response values for the next step are calculated only from quantities belonging to the current step. On the other hand, an implicit scheme is one in which the expressions giving the values for the next step include one or

more values of the next step, and therefore successive iterations are needed to arrive to the solution for the next step. Implicit methods lead in general to increased computational effort, although it is possible for some of them to be converted into an explicit formulation. Algorithms that require two or more implicit systems to be solved simultaneously at each time step have improved properties (Argyris et al. 1973), but they require twice or more the computational effort of the corresponding simple implicit systems.

Another classification that can be made is according to the formulation used to ensure conservation (or decay) of energy within a time step which is a sufficient condition for algorithmic stability (Kuhl and Crisfield 1999). This energy criterion is summarized in the following inequality:

$$(U_{n+1} + K_{n+1}) - (U_n + K_n) \leq W_{\text{ext}} \quad (3)$$

where U_n and U_{n+1} represent the strain energy at the beginning and at the end of the time step respectively, K_n and K_{n+1} are the corresponding kinetic energies and W_{ext} represents the work done by external forces within the time step. This classification results in the following three categories of algorithms which satisfy inequality (3):

- Algorithms which employ numerical dissipation. The algorithms considered in the present study fall in this category.
- Algorithms extending others by using constraints of energy conservation imposed via Lagrange multipliers (Constraint Energy Method), the first of which was presented by Hughes et al. (1978).
- Algorithms which enforce energy conservation algorithmically such as the energy-momentum method presented by Kuhl and Ramm (1996). One of the first algorithms of this category was presented by Simo and Tarnow (1992). In the absence of external loading these algorithms are designed to obey the following laws:

$$\frac{d\mathbf{L}_t}{dt} = 0, \quad \frac{d\mathbf{J}_t}{dt} = 0, \quad \frac{dE_t^{\text{tot}}}{dt} \leq 0 \quad (4)$$

where $\mathbf{L}_t$ is the linear momentum, $\mathbf{J}_t$ is the angular momentum and E_t^{tot} is the total energy. Combinations of algorithms of different categories from the above have also been made. For example, combinations of numerical dissipation algorithms and algorithms ensuring energy conservation algorithmically are presented in Kuhl and Crisfield (1999) and Armero and Petöcz (1996).

Due to the lack of a general time integration algorithm suitable for various complex nonlinear dynamic structural systems, attempts have been made to combine two or more known algorithms into new more efficient ones. This concept of composition is another alternative for the construction of time integration schemes which possess any desired properties, depending on the properties of the parent algorithms being combined. Algorithm composition can be done in two main ways:

- Each time step is divided into two or more substeps, at which different independent integration schemes are applied. Equilibrium is satisfied at each time substep. The final solution depends on the algorithms used as well as on the way of partition of the time steps. The most representative method is presented by Bathe (2007), whereas a classification of these methods is presented by Zhang et al. (2014).
- Different difference formulae are combined in one whole time step. A representative algorithm of this category is presented by Rezaiee-Pajand et al. (2011).

An additional method to solve time dependent problems is the Whole Element Method (WEM) in which time is incorporated along with the other spatial variables into a direct variational method. Therefore, the initial condition problem can be converted into a boundary value problem, containing both spatial and temporal variables (Rosales and Filipich 2002).

Finally, various advanced methods have been successfully employed for the calculation of the dynamic response. Real-time hybrid simulation (RTHS), used by Zhu et al. (2016), combines physical experimentation with numerical simulation to evaluate the dynamic response of structures.

In this study, after a concise literature review of the numerical direct time integration algorithms applicable to the dynamic equilibrium equations of structural analysis of the form (1), a modification of a group of general single step single solve algorithms is presented, which accounts for nonlinear dynamic response. This group of algorithms has already been published in the literature (Zhou and Tamma 2004), but only for the case of linear dynamic response. The modification consists of introducing a Newton–Raphson iterative procedure inside each increment (or time step). Afterwards, the modified algorithm is applied for the direct time integration of the equation of motion [in the form of Eq. (1)] of nonlinear dynamic systems the details of which can be found in Sect. 3. The characteristics of the various integration schemes used in relation to the resulting dynamic response is studied and their relative performances are compared. Apart from its robustness in solving nonlinear problems, it is proved that the extended algorithm can be designed to cope with cases with any degree of nonlinearity.

2 Modified nonlinear time integration algorithm

2.1 The linear generalized single step single solve algorithm

The equation of motion of any multi degree of freedom (MDOF) linear structure is given by the combination of Eqs. (1) and (2):

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \quad (5)$$

with initial conditions:

$$\mathbf{u}(0) = \mathbf{u}_0, \quad \dot{\mathbf{u}}(0) = \dot{\mathbf{u}}_0 \quad (6)$$

Zhou and Tamma (2004) studied a family of general single step single solve (GSSSS) algorithms, namely algorithms which do not involve matrix multiplications and involve only one single system solve in a single time step. A single size of the system refers to the number of degrees-of-freedom resulting from its spatial discretization. It is shown that the Dahlquist theorem holds not only for the linear multistep methods (LMS), but also for the general single step single solve (GSSSS) time integration algorithms, which are spectrally identical to the former. This theorem states that a GSSSS algorithm which is unconditionally stable, can be at most second order accurate.

Equation (5) can be represented as a time weighted residual as follows:

$$\int_{t_n}^{t_{n+1}} W(\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} - \mathbf{f})dt \quad (7)$$

where the weighted time field is assumed to be of the form

$$W = 1 + w_1\Gamma + w_2\Gamma^2 + w_3\Gamma^3 \quad (8)$$

and

$$\Gamma = \bar{\tau}/\Delta t, \quad \bar{\tau} = t - t_n, \quad \Delta t = t_{n+1} - t_n \quad (9)$$

The dependent field variables ($\mathbf{u}$, $\dot{\mathbf{u}}$, $\ddot{\mathbf{u}}$) can be approximated by the following asymptotic series expansions:

$$\mathbf{u} = \mathbf{u}_n + A_1\dot{\mathbf{u}}_n\tau + A_2\ddot{\mathbf{u}}_n\tau^2 + A_3\frac{\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n}{\Delta t}\tau^3 \quad (10)$$

$$\dot{\mathbf{u}} = \dot{\mathbf{u}}_n + A_4\ddot{\mathbf{u}}_n\tau + A_5\frac{\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n}{\Delta t}\tau^2 \quad (11)$$

$$\ddot{\mathbf{u}} = \ddot{\mathbf{u}}_n + A_6\frac{\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n}{\Delta t}\tau \quad (12)$$

and the load vector is expanded to first order via a Taylor series:

$$\mathbf{f} = \mathbf{f}_n + \frac{\mathbf{f}_{n+1} - \mathbf{f}_n}{\Delta t}\tau \quad (13)$$

The updates of displacement and velocity are given by the equations:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1\dot{\mathbf{u}}_n\Delta t + \lambda_2\ddot{\mathbf{u}}_n\Delta t^2 + \lambda_3(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t^2 \quad (14)$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4\ddot{\mathbf{u}}_n\Delta t + \lambda_5(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t \quad (15)$$

The update of acceleration is given by substitution of Eqs. (8) to (15) into (7) as follows:

$$\begin{aligned} & (\mu_6\mathbf{M} + \mu_5\mathbf{C}\Delta t + \mu_3\mathbf{K}\Delta t^2)\ddot{\mathbf{u}}_{n+1} \\ & = -\mathbf{M}(\ddot{\mathbf{u}}_n - \mu_6\ddot{\mathbf{u}}_n) \\ & \quad - \mathbf{C}(\dot{\mathbf{u}}_n + \mu_4\ddot{\mathbf{u}}_n\Delta t - \mu_5\ddot{\mathbf{u}}_n\Delta t) \\ & \quad - \mathbf{K}(\mathbf{u}_n + \mu_1\dot{\mathbf{u}}_n\Delta t + \mu_2\ddot{\mathbf{u}}_n\Delta t^2 - \mu_3\ddot{\mathbf{u}}_n\Delta t^2) \\ & \quad + (1 - W_1)\mathbf{f}_n + W_1\mathbf{f}_{n+1} \end{aligned} \quad (16)$$

or in a simplified form:

$$\tilde{\mathbf{M}}\ddot{\mathbf{u}}_{n+1} = \tilde{\mathbf{F}}_n \quad (17)$$

where

$$\begin{aligned} \tilde{\mathbf{F}}_n(\mathbf{K}, \mathbf{C}, \mathbf{f}) & = -\mathbf{M}(\ddot{\mathbf{u}}_n - \mu_6\ddot{\mathbf{u}}_n) - \mathbf{C}(\dot{\mathbf{u}}_n + \mu_4\ddot{\mathbf{u}}_n\Delta t - \mu_5\ddot{\mathbf{u}}_n\Delta t) \\ & \quad - \mathbf{K}(\mathbf{u}_n + \mu_1\dot{\mathbf{u}}_n\Delta t + \mu_2\ddot{\mathbf{u}}_n\Delta t^2 - \mu_3\ddot{\mathbf{u}}_n\Delta t^2) + (1 - W_1)\mathbf{f}_n + W_1\mathbf{f}_{n+1} \end{aligned} \quad (18)$$

and

$$\tilde{\mathbf{M}}(\mathbf{K}, \mathbf{C}) = \mu_6\mathbf{M} + \mu_5\mathbf{C}\Delta t + \mu_3\mathbf{K}\Delta t^2 \quad (19)$$

The subscript of $\tilde{\mathbf{F}}_n$ denotes the time step at which the quantities $\mathbf{u}$, $\dot{\mathbf{u}}$, $\ddot{\mathbf{u}}$ are evaluated for its calculation. The time step at which $\mathbf{K}$, $\mathbf{C}$ and $\mathbf{f}$ are evaluated is denoted by a separate subscript placed at these quantities. The constants W_i are given by:

$$W_i = \frac{\frac{1}{1+i} + \sum_{j=1}^3 \frac{w_j}{1+i+j}}{1 + \sum_{j=1}^3 \frac{w_j}{1+j}}, \quad i = 1, 2, 3 \quad (20)$$

There are 12 independent integration constants that are needed in order to apply the Eqs. (14), (15) and (16) to proceed to the next step. These are $W_1, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$. Each set of these parameters defines a unique algorithm, and can be considered in some way as the algorithm's signature. Many known time integration algorithms, which will be presented later, result from suitable selection of these parameters. Zhou and Tamma (2004) calculate the integration parameters by imposing several different constraints to the algorithm, regarding the order of accuracy, the overshooting behavior (in terms of displacement and velocity orders), the spurious roots at the high and low frequency limits, the dissipation and dispersion properties, the bifurcation of the principal roots, etc., which results in the derivation of 9 different algorithms belonging to the above family. In this study, W_1 is calculated directly from (20), after specifying the parameters w_1, w_2, w_3 , so the number of integration constants needed are 14.

2.2 Design of the linear generalized single step single solve algorithm: special cases

An algorithm is termed to have the property of continuous acceleration, if, at the low frequency limit, the acceleration $\ddot{\mathbf{u}}_{n+1}$ calculated at $t = t_n$ satisfies the equation of motion (strong form) at $t = t_n$. Otherwise, the algorithm is termed to have the property of discontinuous acceleration (Zhou and Tamma 2004).

The procedure for designing the algorithm presented in the previous section to apply it to time integration problems (i.e. setting its 14 integration constants), is presented in Zhou and Tamma (2004). The algorithms of the generalized single step family are shown in Table 1.

The values of the integration constants are shown for various known integration schemes in tables. In Table 2 the parameters of the central difference method, the general family of Newmark methods (Newmark 1959), the Average Constant Acceleration method (Newmark 1959), the Linear Acceleration method (Newmark 1959), the Backward Acceleration Method (Ascher and Petzold 1998), and the Fox–Goodwin formula (Fox and Goodwin 1949) are shown. In the case of the general family of Newmark methods, β and γ

Table 1 Algorithms of the generalized single step single solve (GSSSS) family

Method name	Displacement overshooting order	Velocity overshooting order	Type
U0-V0-Opt	0	0	Optimal numerical dissipation and dispersion
U0-V0-CA	0	0	Continuous acceleration
U0-V0-DA	0	0	Discontinuous acceleration
U0-V1-Opt	0	1	Optimal numerical dissipation and dispersion
U0-V1-CA	0	1	Continuous acceleration
U0-V1-DA	0	1	Discontinuous acceleration
U1-V0-Opt	1	0	Optimal numerical dissipation and dispersion
U1-V0-CA	1	0	Continuous acceleration
U1-V0-DA	1	0	Discontinuous acceleration

Table 2 Integration parameters for various known time integration schemes

	Central difference method	Family of Newmark methods (Newmark 1959)	Newmark average constant acceleration <1> (Newmark 1959)	Newmark linear acceleration <2> (Newmark 1959)	Newmark backward acceleration <3> (Ascher and Petzold 1998)	Fox–Goodwin formula <4> (Fox and Goodwin 1949)
w_1	−15	−15	−15	−15	−15	−15
w_2	45	45	45	45	45	45
w_3	−35	−35	−35	−35	−35	−35
μ_1	1	1	1	1	1	1
μ_2	0	β	−	$1/6$	$1/2$	$1/12$
μ_3	0	β	$1/4$	$1/6$	$1/2$	$1/12$
μ_4	$1/2$	γ	$1/2$	$1/2$	$1/2$	$1/2$
μ_5	$1/2$	γ	$1/2$	$1/2$	$1/2$	$1/2$
μ_6	1	1	1	1	1	1
λ_1	1	1	1	1	1	1
λ_2	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$	$1/2$
λ_3	0	β	$1/4$	$1/6$	$1/2$	$1/12$
λ_4	1	1	1	1	1	1
λ_5	$1/2$	γ	$1/2$	$1/2$	$1/2$	$1/2$

are the well-known Newmark constants. In Table 3 the parameters are given for the zero-order displacement, first-order velocity overshooting algorithms, presented in (Zhou and Tamma 2004). It has to be mentioned that the formulas presented in Table 3 correspond to three special cases of these zero-order displacement, first-order velocity overshooting algorithms, namely the generalized α -method, the HHT- α method and the WBZ α -method, presented by Chung and Hulbert (1993), Hilber et al. (1977), Wood et al. (1980) respectively. In order to evaluate the integration constants, the spectral radius ρ_∞ , which is the minimum absolute value of the principal roots of the amplification matrix at the high-frequency limit, has to be first assigned a desired value, which must lie in the range given at the first row of Tables 3, 4 and 5. If $\rho_\infty = 1$, the resulting algorithm is non-dissipative. In Table 4 the parameters of the zero-order displacement and zero-order velocity overshooting algorithms are presented (Zhou and Tamma 2004). Table 5 shows the parameters of the first-order displacement and zero-order velocity overshooting algorithms. If $\rho_\infty = 1$, the first-order displacement, zero-order velocity, optimal numerical dissipation and dispersion, the first-order displacement, zero-order velocity, continuous acceleration and the first-order displacement, zero-order velocity, discontinuous acceleration algorithms recover, the first the mid-point rule α -form algorithm, and the two last the Newmark average acceleration α -form algorithm. In Tables 2, 3, 4 and 5, the number inside the angle brackets after the name of each algorithm is the number with which the algorithm is referred to in this study. This is done merely for easy identification of each algorithm throughout the paper.

Table 3 Integration parameters for zero-order displacement, first order velocity overshooting algorithms

	U_0-V_1 -Opt $\langle 5 \rangle$, (Chung and Hulbert 1993)	U_0-V_1 -CA $\langle 6 \rangle$, (Hilber et al. 1977)	U_0-V_1 -DA $\langle 7 \rangle$, (Wood et al. 1980)
ρ_∞	[0 1]	[1/2 1]	[0 1]
w_1	$-15 \frac{1-2\rho_\infty}{1-4\rho_\infty}$	$-15 \frac{1-2\rho_\infty}{2-3\rho_\infty}$	-15
w_2	$15 \frac{3-4\rho_\infty}{1-4\rho_\infty}$	$15 \frac{2-5\rho_\infty}{2-3\rho_\infty}$	45
w_3	$-35 \frac{1-\rho_\infty}{1-4\rho_\infty}$	$-35 \frac{1-3\rho_\infty}{2(2-3\rho_\infty)}$	-35
μ_1	$\frac{1}{1+\rho_\infty}$	$\frac{2\rho_\infty}{1+\rho_\infty}$	1
μ_2	$\frac{1}{2(1+\rho_\infty)}$	$\frac{\rho_\infty}{1+\rho_\infty}$	$\frac{1}{2}$
μ_3	$\frac{1}{(1+\rho_\infty)^3}$	$\frac{2\rho_\infty}{(1+\rho_\infty)^3}$	$\frac{1}{(1+\rho_\infty)^2}$
μ_4	$\frac{1}{1+\rho_\infty}$	$\frac{2\rho_\infty}{1+\rho_\infty}$	1
μ_5	$\frac{3-\rho_\infty}{2(1+\rho_\infty)^2}$	$\frac{\rho_\infty(3-\rho_\infty)}{(1+\rho_\infty)^2}$	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$
μ_6	$\frac{2-\rho_\infty}{1+\rho_\infty}$	1	$\frac{2}{1+\rho_\infty}$
λ_1	1	1	1
λ_2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
λ_3	$\frac{1}{(1+\rho_\infty)^2}$	$\frac{1}{(1+\rho_\infty)^2}$	$\frac{1}{(1+\rho_\infty)^2}$
λ_4	1	1	1
λ_5	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$

Table 4 Integration parameters for zero-order displacement, zero order velocity overshooting algorithms

	U_0-V_0 -Opt $\langle 8 \rangle$	U_0-V_0 -CA $\langle 9 \rangle$	U_0-V_0 -DA $\langle 10 \rangle$
ρ_∞	[0 1]	[1/3 1]	[0 1]
w_1	$-15 \frac{1-2\rho_\infty}{1-4\rho_\infty}$	$-15 \frac{1-5\rho_\infty}{3-7\rho_\infty}$	-15
w_2	$15 \frac{3-4\rho_\infty}{1-4\rho_\infty}$	$15 \frac{1-13\rho_\infty}{3-7\rho_\infty}$	45
w_3	$-35 \frac{1-\rho_\infty}{1-4\rho_\infty}$	$140 \frac{\rho_\infty}{3-7\rho_\infty}$	-35
μ_1	$\frac{1}{1+\rho_\infty}$	$\frac{1+3\rho_\infty}{2(1+\rho_\infty)}$	1
μ_2	$\frac{1}{2(1+\rho_\infty)}$	$\frac{1+3\rho_\infty}{4(1+\rho_\infty)}$	$\frac{1}{2}$
μ_3	$\frac{1}{2(1+\rho_\infty)^2}$	$\frac{1+3\rho_\infty}{4(1+\rho_\infty)^2}$	$\frac{1}{2(1+\rho_\infty)}$
μ_4	$\frac{1}{1+\rho_\infty}$	$\frac{1+3\rho_\infty}{2(1+\rho_\infty)}$	1
μ_5	$\frac{1}{(1+\rho_\infty)^2}$	$\frac{1+3\rho_\infty}{2(1+\rho_\infty)^2}$	$\frac{1}{1+\rho_\infty}$
μ_6	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$	1	$\frac{3+\rho_\infty}{2(1+\rho_\infty)}$
λ_1	1	1	1
λ_2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
λ_3	$\frac{1}{2(1+\rho_\infty)}$	$\frac{1}{2(1+\rho_\infty)}$	$\frac{1}{2(1+\rho_\infty)}$
λ_4	1	1	1
λ_5	$\frac{1}{1+\rho_\infty}$	$\frac{1}{1+\rho_\infty}$	$\frac{1}{1+\rho_\infty}$

Table 5 Integration parameters for first-order displacement, zero order velocity overshooting algorithms

	U ₁ -V ₀ -Opt <11>	U ₁ -V ₀ -CA <12>	U ₁ -V ₀ -DA <13>
ρ_∞	[0 1]	[1/2 1]	[0 1]
w_1	$-30 \frac{3-8\rho_\infty+6\rho_\infty^2}{9-22\rho_\infty+19\rho_\infty^2}$	$-60 \frac{2-8\rho_\infty+7\rho_\infty^2}{11-48\rho_\infty+41\rho_\infty^2}$	$-30 \frac{3-4\rho_\infty}{9-11\rho_\infty}$
w_2	$-15 \frac{25-74\rho_\infty+53\rho_\infty^2}{2(9-22\rho_\infty+19\rho_\infty^2)}$	$15 \frac{37-140\rho_\infty+127\rho_\infty^2}{2(11-48\rho_\infty+41\rho_\infty^2)}$	$15 \frac{25-37\rho_\infty}{2(9-11\rho_\infty)}$
w_3	$-35 \frac{3-10\rho_\infty+7\rho_\infty^2}{9-22\rho_\infty+19\rho_\infty^2}$	$-35 \frac{5-18\rho_\infty+17\rho_\infty^2}{11-48\rho_\infty+41\rho_\infty^2}$	$-35 \frac{3-5\rho_\infty}{9-11\rho_\infty}$
μ_1	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$	$\frac{1+3\rho_\infty}{2(1+\rho_\infty)}$	$\frac{3+\rho_\infty}{2(1+\rho_\infty)}$
μ_2	$\frac{1}{(1+\rho_\infty)^2}$	$\frac{2\rho_\infty}{(1+\rho_\infty)^2}$	$\frac{1}{1+\rho_\infty}$
μ_3	$\frac{1}{(1+\rho_\infty)^3}$	$\frac{2\rho_\infty}{(1+\rho_\infty)^3}$	$\frac{1}{(1+\rho_\infty)^2}$
μ_4	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$	$\frac{1+3\rho_\infty}{2(1+\rho_\infty)}$	$\frac{3+\rho_\infty}{2(1+\rho_\infty)}$
μ_5	$\frac{2}{(1+\rho_\infty)^3}$	$\frac{4\rho_\infty}{(1+\rho_\infty)^3}$	$\frac{2}{(1+\rho_\infty)^2}$
μ_6	$\frac{2-\rho_\infty}{1+\rho_\infty}$	1	$\frac{2}{1+\rho_\infty}$
λ_1	1	1	1
λ_2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
λ_3	$\frac{1}{2(1+\rho_\infty)}$	$\frac{1}{2(1+\rho_\infty)}$	$\frac{1}{(1+\rho_\infty)^2}$
λ_4	1	1	1
λ_5	$\frac{1}{1+\rho_\infty}$	$\frac{1}{1+\rho_\infty}$	$\frac{3-\rho_\infty}{2(1+\rho_\infty)}$

2.3 Modification of the linear algorithm for nonlinear dynamic response

In this section the family of generalized linear algorithms presented above is modified to account for materially nonlinear dynamic response. In general, to proceed from the current step ($\mathbf{u}_n, \dot{\mathbf{u}}_n, \ddot{\mathbf{u}}_n$) to the next time step ($\mathbf{u}_{n+1}, \dot{\mathbf{u}}_{n+1}, \ddot{\mathbf{u}}_{n+1}$), the secant stiffness and damping matrices are needed, which usually depend on $\mathbf{u}_{n+1}$ and $\dot{\mathbf{u}}_{n+1}$. Since the latter are unknown, the tangent stiffness and damping matrices are calculated and iterations are performed to arrive to a converged solution. Convergence is attained via a Newton–Raphson iterative procedure. In some time integration algorithms, this iteration is avoided by using the initial tangent matrices instead of updating them, even though this approximation is not correct in principle.

The outline of the modified nonlinear time integration algorithm used in this study is shown in Fig. 1. The given data are the mass, stiffness and damping properties of the system and the imposed external force, denoted by $\mathbf{M}, \bar{\mathbf{K}}, \bar{\mathbf{C}}, \mathbf{f}$ respectively. Note that the various quantities correspond to a MDOF system in general; this means that $\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}, \mathbf{f}$ are column vectors, and $\mathbf{M}, \bar{\mathbf{K}}, \bar{\mathbf{C}}$ are square matrices (vectors and matrices are denoted with bold characters throughout the text). The symbol./denotes right array division, namely the division of two vectors of the same size in an element by element fashion. However, if SDOF systems are considered, all the above quantities degenerate into scalar quantities.

Before the application of the algorithm, the necessary integration constants are calculated and the maximum tolerance $tol_{\max}$ and the maximum number of iterations until convergence $k_{\max}$ are set.

```

Set  $w_1, w_2, w_3, W_1\Lambda_1, W_2\Lambda_2, W_3\Lambda_3, W_4\Lambda_4, W_5\Lambda_5, W_6\Lambda_6, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ 
Find  $W_i$  from (20) for  $i=1$ 
Initialize  $u_n = u_0, \dot{u}_n = \dot{u}_0$ 
Find  $K_0 = \bar{K}(u_0, \dot{u}_0), C_0 = \bar{C}(u_0, \dot{u}_0), p_0 = \bar{p}(u_0, \dot{u}_0)$ 
Find  $\ddot{u}_0 = (f_0 - p_0)/M$  from (1)
Set  $K_n = K_0, C_n = C_0, p_n = p_0, u_n = u_0, \dot{u}_n = \dot{u}_0, \ddot{u}_n = \ddot{u}_0$ 
for  $n = 1:\text{length}(f)-1$ 
    Initialize  $k = 1$ 
    Initialize  $\text{tol} = \text{tol}_{\max}$ 
    Find  $\ddot{u}_{n+1}^1 = \tilde{M}^{-1}(K_n, C_n) \tilde{F}_n(K_n, C_n, p_n)$  from (17)
    while  $\text{tol} \geq \text{tol}_{\max} \ \& \ k \leq k_{\max}$ 
        Iteration  $k$  of increment  $n+1$ :
        Set  $\ddot{u}_{n+1}^k = \ddot{u}_n + \ddot{u}_{n+1}^1$ 
        Find  $u_{n+1}^k$  and  $\dot{u}_{n+1}^k$  according to (14) and (15) respectively
        Find  $K_{n+1}^k = \bar{K}(u_{n+1}^k, \dot{u}_{n+1}^k), C_{n+1}^k = \bar{C}(u_{n+1}^k, \dot{u}_{n+1}^k), p_{n+1}^k = \bar{p}(u_{n+1}^k, \dot{u}_{n+1}^k)$ 
        Find the residual  $R_{n+1}^k = \tilde{F}_{n+1}^k(K_{n+1}^k, C_{n+1}^k, p_{n+1}^k) - \tilde{M}(K_{n+1}^k, C_{n+1}^k) \ddot{u}_{n+1}^1$ 
        Set  $d\ddot{u}_{n+1}^k = \ddot{u}_n + (\ddot{u}_{n+1}^1 + da)$ , where  $da$  is a constant infinitesimal variation of
        acceleration
        Find  $du_{n+1}^k$  and  $d\dot{u}_{n+1}^k$  from (14) and (15) respectively
        Find  $dK_{n+1}^k = \bar{K}(du_{n+1}^k, d\dot{u}_{n+1}^k), dC_{n+1}^k = \bar{C}(du_{n+1}^k, d\dot{u}_{n+1}^k), dp_{n+1}^k = \bar{p}(du_{n+1}^k, d\dot{u}_{n+1}^k)$ 
        Find the residual  $dR_{n+1}^k = d\tilde{F}_{n+1}^k(dK_{n+1}^k, dC_{n+1}^k, dp_{n+1}^k) - d\tilde{M}(dK_{n+1}^k, dC_{n+1}^k) d\ddot{u}_{n+1}^k$ 
        Find  $qda = [R_{n+1}^k / \ddot{u}_{n+1}^1] / \sqrt{[dR_{n+1}^k - R_{n+1}^k] / da}$ 
        Update  $\ddot{u}_{n+1}^1 = (1 - qda) \ddot{u}_{n+1}^1$ 
        Update  $k = k + 1$ 
    end
    Set  $\ddot{u}_{n+1}^k = \ddot{u}_n + \ddot{u}_{n+1}^1$ 
    Find  $u_{n+1}^k$  and  $\dot{u}_{n+1}^k$  according to (14) and (15) respectively
    Find  $K_{n+1}^k = \bar{K}(u_{n+1}^k, \dot{u}_{n+1}^k), C_{n+1}^k = \bar{C}(u_{n+1}^k, \dot{u}_{n+1}^k), p_{n+1}^k = \bar{p}(u_{n+1}^k, \dot{u}_{n+1}^k)$ 
    Assign for next increment:  $K_{n+1} = K_{n+1}^k, C_{n+1} = C_{n+1}^k, p_{n+1} = p_{n+1}^k, u_{n+1} = u_{n+1}^k, \dot{u}_{n+1} = \dot{u}_{n+1}^k,$ 
     $\ddot{u}_{n+1} = \ddot{u}_{n+1}^k$ 
end
    
```

Fig. 1 Flowchart of the nonlinear algorithms used in this study

3 Benchmark SDOF systems for evaluation of algorithms' performance

In this section, 13 different time integration schemes presented in the Tables 2, 3, 4 and 5 are compared through their application for solving a number of elementary (benchmark) problems. The schemes compared are the Newmark Average Constant Acceleration method (Newmark 1959), denoted by <1>, the Newmark Linear Acceleration method (Newmark 1959), denoted by <2>, the Newmark Backward Acceleration method (Ascher and Petzold 1998), denoted by <3>, the Fox–Goodwin formula (Fox and Goodwin 1949), denoted by <4>, the U0-V1-Opt, denoted by <5>, the U0-V1-CA, denoted by <6>, the

U0-V1-DA, denoted by <7>, the U0-V0-Opt, denoted by <8>, the U0-V0-CA, denoted by <9>, the U0-V0-DA, denoted by <10>, the U1-V0-Opt, denoted by <11>, the U1-V0-CA, denoted by <12>, and the U1-V0-DA, denoted by <13>, algorithms. The last 9 integration schemes are presented by Zhou and Tamma (2004) and details about their notation can be seen in Tables 3, 4, 5.

A total of seven benchmark problems have been considered in this study, involving SDOF oscillators with various constitutive relations, damping types, the dynamic response of which occurs purely due to nonzero initial conditions (unforced), as well as a linear 3DOF system excited through an imposed displacement. The last has been studied by Bathe and Noh (2012) and its benchmark value is that it represents the stiff and flexible components of a complex structural system. The aforementioned examples have been studied in order to assess the performance of the various time integration schemes. Their description will be made in the following sections. In each problem, the ρ_{inf} parameter is selected to be equal to zero, or the lowest possible value for all integration algorithms used. Concerning the Newton–Raphson iterative procedure used, the maximum convergence tolerance and the maximum number of iterations are $\text{tol}_{\text{max}} = 0.01$ and $k_{\text{max}} = 200$ respectively. All units involved in the calculations belong to the SI system. For simplification of the calculations, the mass of all the SDOF systems considered is set to unity, without loss of generality. The time step used is $\Delta t = 0.01$ and the duration of the dynamic analysis is equal to 100 time steps (1 s) for all problems.

3.1 Undamped SDOF oscillator with hardening spring

The first benchmark problem studied is a SDOF oscillator with hardening spring, for which the equation of motion is:

$$m\ddot{u} + S_1 u(1 + S_2 u^2) = 0 \quad (21)$$

This type of oscillator represents the well-known unforced and undamped Duffing oscillator. The system is conservative and its total energy is constant and given by analytical integration of (21):

$$E_{\text{tot}} = \frac{1}{2} m \dot{u}^2 + \frac{1}{2} S_1 u^2 + \frac{1}{4} S_1 S_2 u^4 \quad (22)$$

Oscillators of such type have been examined by Xie (1996) and Liu and Wang (2008). In this example $S_1 = 1000$, $S_2 = 0.1$ and the initial conditions are $u_0 = 1.5$ and $\dot{u}_0 = 0$.

3.2 Undamped SDOF oscillator with softening spring

The second benchmark problem studied is a SDOF oscillator with softening spring, for which the equation of motion is:

$$m\ddot{u} + S \tanh(u) = 0 \quad (23)$$

The system is conservative and its total energy is constant and given by analytical integration of (23):

$$E_{\text{tot}} = \frac{1}{2}m\dot{u}^2 + S \ln(\cosh(u)) \quad (24)$$

Oscillators of such type have been examined by Xie (1996) and Liu and Wang (2008). In this example $S = 1000$ and the initial conditions are $u_0 = 0.3$ and $\dot{u}_0 = 0$.

3.3 Undamped SDOF oscillator with linear elastic spring

The third benchmark problem studied is a SDOF oscillator with linear elastic spring, for which the equation of motion is:

$$m\ddot{u} + ku = 0 \quad (25)$$

The system is conservative and its total energy is constant and given by analytical integration of (25):

$$E_{\text{tot}} = \frac{1}{2}m\dot{u}^2 + \frac{1}{2}ku^2 \quad (26)$$

In this example $k = 1000$ and the initial conditions are $u_0 = 4$ and $\dot{u}_0 = 0$.

3.4 SDOF oscillator with linear elastic spring and viscous damping

The fourth benchmark problem studied is a SDOF oscillator with linear elastic spring and viscous damping, for which the equation of motion is:

$$m\ddot{u} + c\dot{u} + ku = 0 \quad (27)$$

The damping coefficient is given by the equation:

$$c = 2\zeta\sqrt{\frac{k}{m}} \quad (28)$$

where ζ is the critical viscous damping ratio equal to 2 %. The system is not conservative; its total energy gradually decreases due to the damping force according to the equation:

$$E_{\text{tot}} = \left(\frac{1}{2}m\dot{u}_0^2 + \frac{1}{2}ku_0^2 \right) \exp\left(\frac{-2\zeta\sqrt{k/m}}{\sqrt{1-\zeta^2}} t \right) \quad (29)$$

In this example $k = 1000$ and the initial conditions are $u_0 = 4$ and $\dot{u}_0 = 0$.

3.5 SDOF oscillator with linear elastic spring and Coulomb damping

The fifth benchmark problem studied is a SDOF oscillator with linear elastic spring and Coulomb damping, for which the equation of motion is:

$$m\ddot{u} + ku + F \frac{\dot{u}}{|\dot{u}|} = 0 \quad (30)$$

The system is not conservative; its total energy gradually decreases due to the Coulomb friction according to the equation:

$$E_{\text{tot}} = \frac{1}{2}k \left(u_0 - 2u_F \frac{\sqrt{k/m}}{\pi} t \right)^2 \quad (31)$$

and $u_F = F/k$ is a constant which may be interpreted as the static deformation of the system due to friction force F . In this example $k = 1000$, $u_F = 0.05$ and the initial conditions are $u_0 = 4$ and $\dot{u}_0 = 0$.

3.6 SDOF oscillator with linear elastic spring and hysteretic damping

The sixth benchmark problem studied is a SDOF oscillator with linear elastic spring and hysteretic damping, for which the equation of motion is:

$$m\ddot{u} + c\dot{u} + ku = 0 \quad (32)$$

The damping coefficient is given by the equation:

$$c = \eta \frac{k}{\omega} \quad (33)$$

where η is the critical hysteretic damping ratio equal to 5 % and ω is the circular eigenfrequency of the SDOF system (which is also its vibration frequency). The system is not conservative; its total energy gradually decreases due to the hysteretic damping force. Analytical equation for the estimation of the energy loss does not exist; the comparison is made with reference to the energy loss associated with the same benchmark problem, calculated for a much smaller time step. In this example $k = 1000$ and the initial conditions are $u_0 = 4$ and $\dot{u}_0 = 0$. The time step used for the reference energy loss is equal to $\Delta\tau = 0.0001$ and the duration of the dynamic analysis in this case is 10,000 time steps.

4 Results of benchmark problems in terms of total energy

Since no energy considerations were made for the formulation of the linear version of the time integration algorithm family assessed in this study, the evaluation of the various algorithms can be made in terms of energy measures, for the problems 1–6. An efficient nonlinear time integration scheme should result in an accurate calculation of the energy of these systems. In cases that the analytical calculation of the energy is possible, the numerically calculated energy is compared to the former; in cases that this calculation is not possible (problem 6), the comparison is made with the energy calculated for a much smaller time step, a result which is practically the same for all algorithms involved. All the numerical analyses involved in this study were performed using MATLAB programming language.

In the diagrams of Figs. 2, 3, 4, 5, 6 and 7, the vertical axis represents the natural logarithm of the total energy of the vibrating system (E_{tot}) and the horizontal axis represents time. To avoid confusion due to the large number of algorithms being studied here, a separate graph has been constructed for each algorithm, and incorporated as a subplot in each figure. The various subplots of these figures have the same scaling in their axes, to enable numerical comparisons by simple observation of the deviation of the various curves from the dashed curve which gives the accurate value of the total energy of each system. The first four graphs in each figure from Figs. 2, 3, 4, 5, 6 and 7, distributed in the two first horizontal rows present results of the elementary time integration schemes considered in

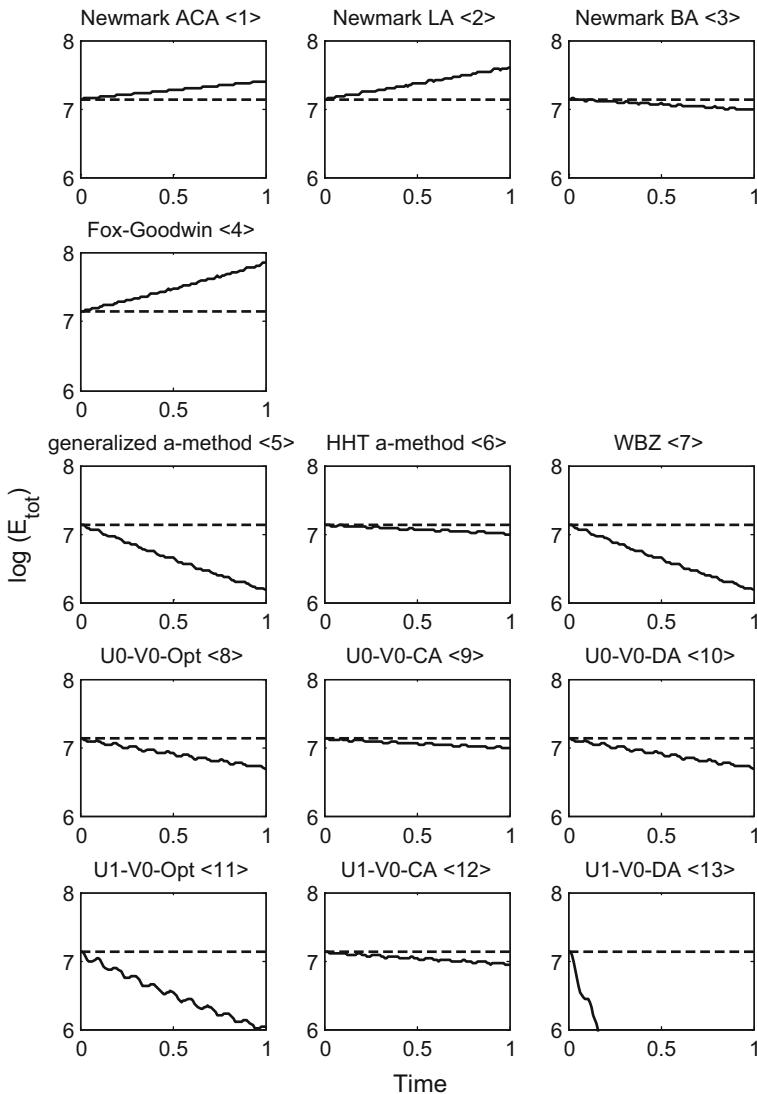


Fig. 2 Natural logarithm of the total energy for problem 1 (undamped SDOF system with hardening spring)

this study (i.e. the Newmark ACA, Newmark LA, Newmark BA and Fox–Goodwin methods), and the last nine graphs in each figure, distributed in the last three rows, present results of the advanced integration schemes (i.e. those belonging to the family of the GSSSS algorithms).

Another point to be highlighted is that, since the size of the time step and the duration of the dynamic analysis is the same for all the benchmark problems except for problem 7 (described in Sect. 5), and the small-strain stiffness and mass of the SDOF systems analyzed is the same, these factors which affect the performance of the time integration schemes are isolated from the comparisons between the results, and the differences

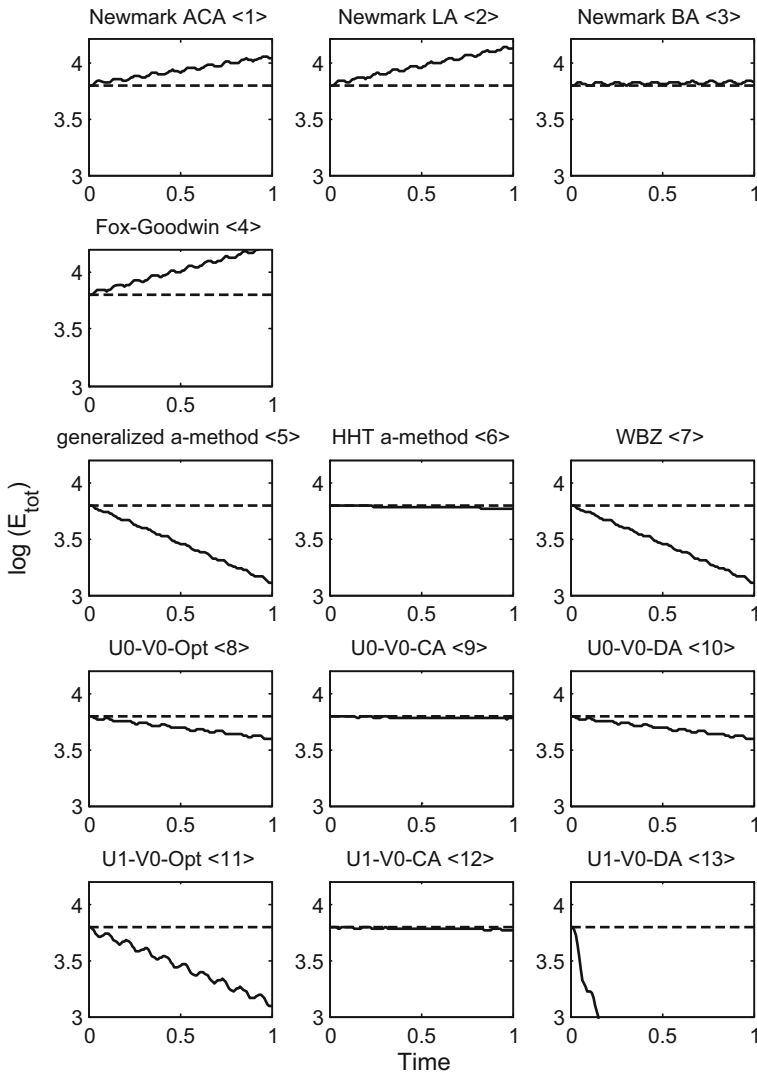


Fig. 3 Natural logarithm of the total energy for problem 2 (undamped SDOF system with softening spring)

observed originate from the nature of the oscillators and the quality of the algorithms (i.e. the magnitude of the numerical error introduced by them). This fact enables the easier understanding of the numerical performance of the time integration schemes considered in this study.

In Fig. 2, the natural logarithm of the total energy of the undamped SDOF system with hardening spring (benchmark problem described in section 3.1) is presented for each algorithm. The exact value of the total energy is calculated at the onset of the vibration, should remain constant throughout the dynamic response, since there is no energy loss due to damping or other sources, and is plotted in all subplots with a dashed line. It is noted that the continuous acceleration (CA) algorithms, along with the Newmark Backward

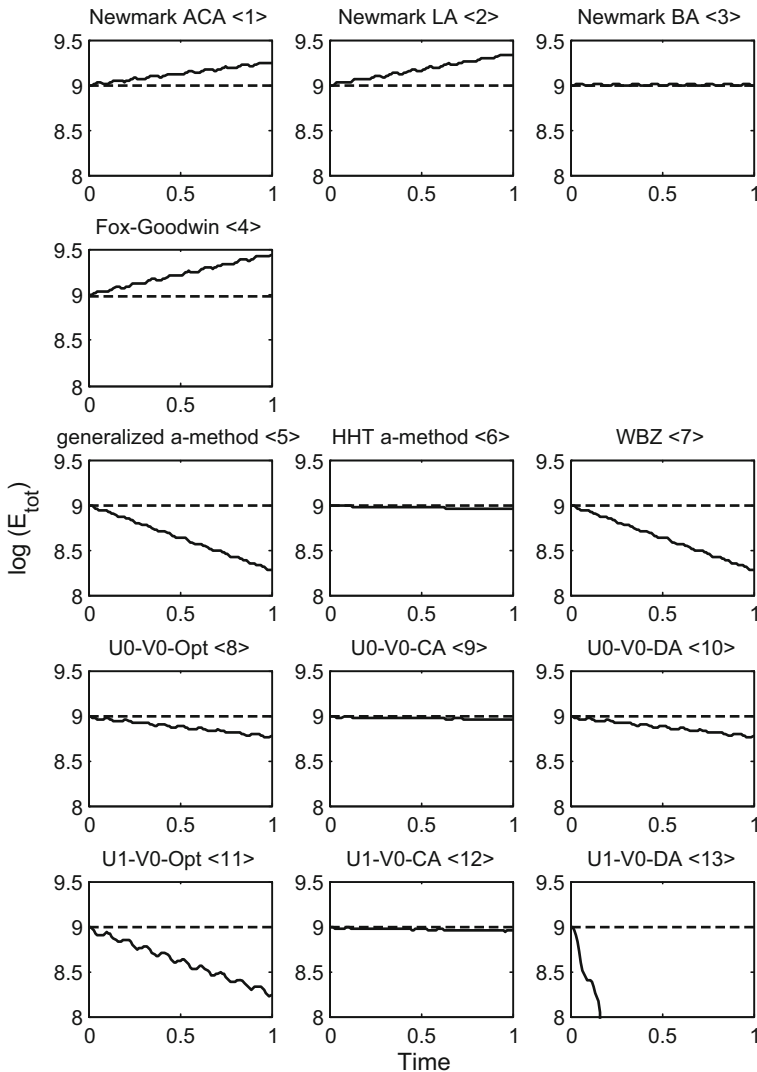


Fig. 4 Natural logarithm of the total energy for problem 3 (undamped SDOF system with linear elastic spring)

Acceleration Method perform generally better than the others in this problem. The least energy error is observed for the HHT-a method (or U0-V1-CA method <6>), i.e. the zero order displacement-first order velocity-continuous acceleration method. Furthermore, the U0-V0-Opt (<8>) and U0-V0-DA (<10>) algorithms seem to perform better than their U0-V1 or U1-V0 counterparts. It was checked that if the size of the time step is sufficiently reduced, all the algorithms of the GSSSS family produce the exact response. The largest numerical error is observed in the case of U1-V0-DA (<13>) algorithm. Finally, it is observed that numerical errors associated with the time integration schemes may result in increase of the total energy in some cases.

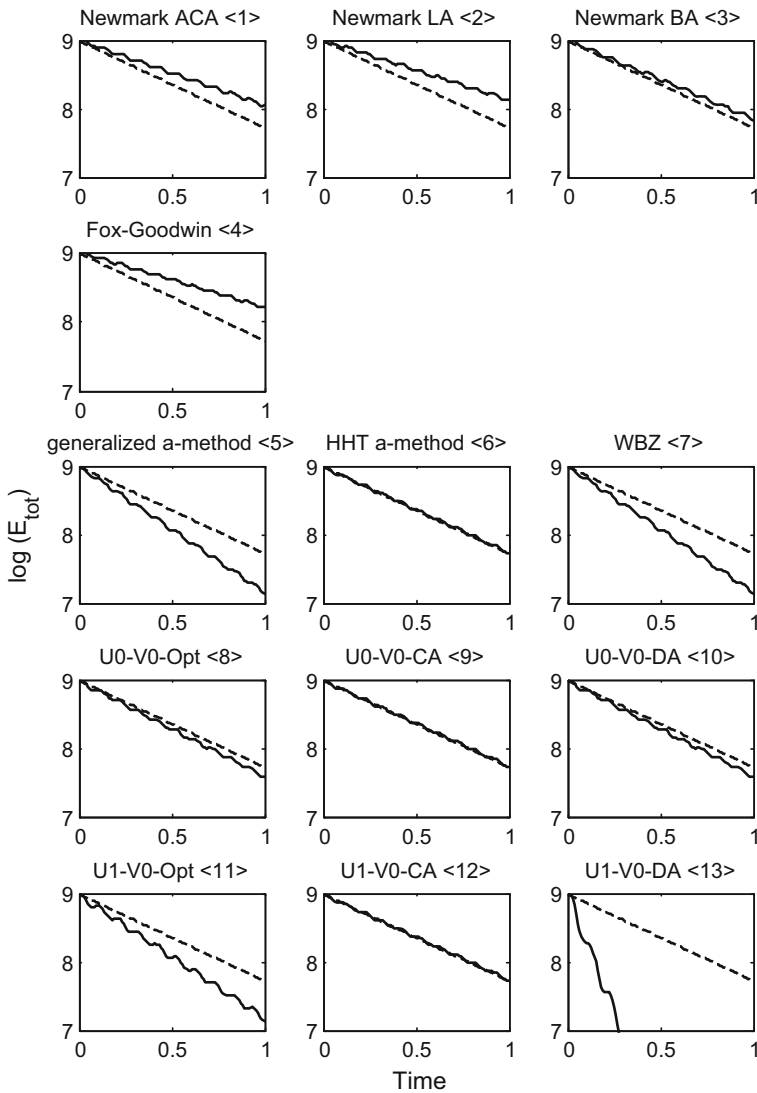


Fig. 5 Natural logarithm of the total energy for problem 4 (linear elastic SDOF system with viscous damping)

In Fig. 3, the natural logarithm of the total energy of the undamped SDOF system with softening spring (benchmark problem described in section 3.2) is presented for each algorithm. Again the most accurate method is the HHT-a method, in the graph of which the difference between the numerically calculated total energy and the analytical reference energy is hard to discern. The superiority of the Continuous Acceleration advanced algorithms is remarkable. The largest error is observed for the case of U1-V0-DA (<13>) algorithm, as in the case of the undamped SDOF system with hardening spring. In general, undulations are exhibited in the energy curves of the less accurate algorithms, either advanced or elementary. The total energy calculated may decrease or increase with respect

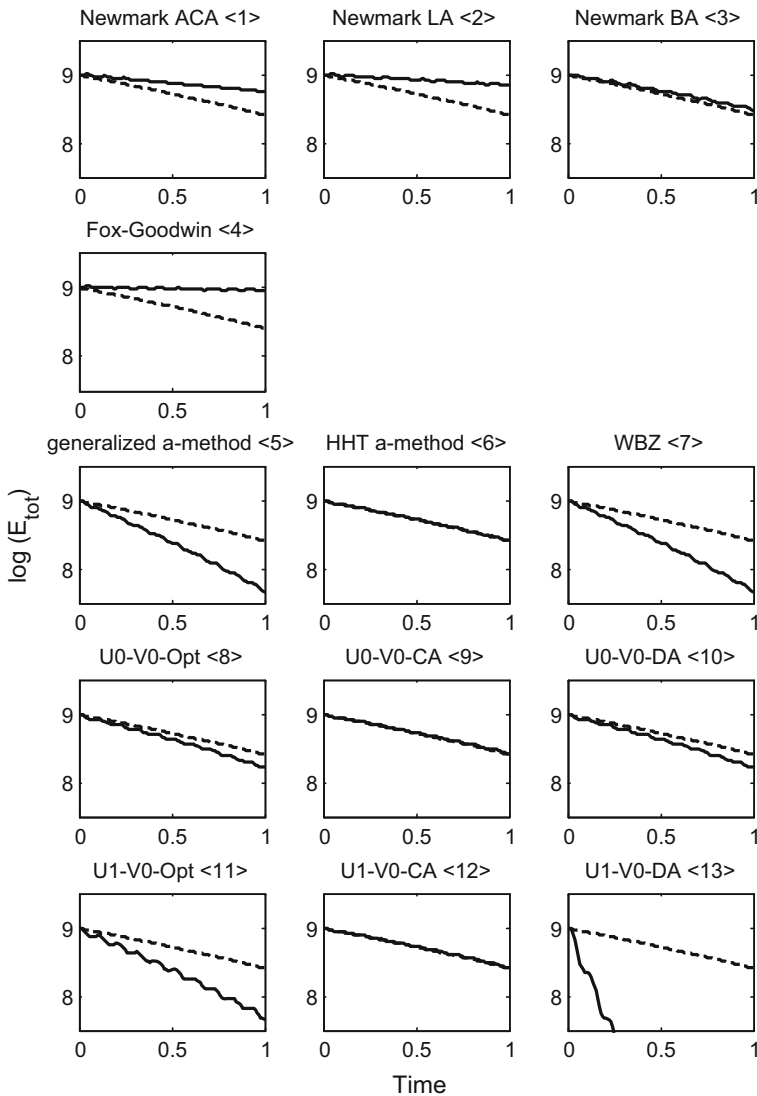


Fig. 6 Natural logarithm of the total energy for problem 5 (linear elastic SDOF system with damping due to Coulomb friction)

to the reference energy curve, depending on the numerical error introduced by each algorithm.

In Fig. 4, the natural logarithm of the total energy of the undamped SDOF system with linear elastic spring (benchmark problem described in section 3.3) is presented. It is seen that the HHT-a method is the most accurate, and among the elementary methods, the Newmark Backward Acceleration method is of comparable accuracy with the HHT-a method. Apart from the Continuous Acceleration and the Newmark BA methods, the

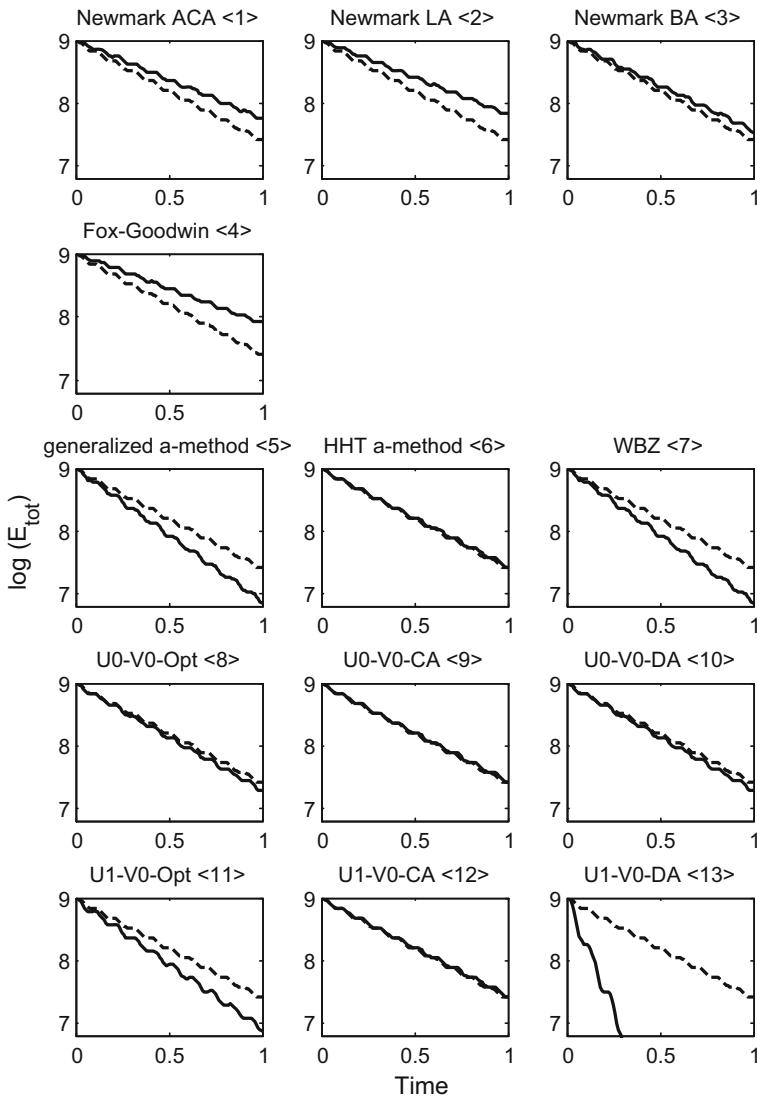


Fig. 7 Natural logarithm of the total energy for problem 6 (linear elastic SDOF system with hysteretic damping)

remaining algorithms give less accurate results, of various orders of error. The least accurate method is the U1-V0-DA (<13>) method.

In Fig. 5, the natural logarithm of the total energy of the linear elastic SDOF system with viscous damping (benchmark problem described in section 3.4) is presented. Similarly to the undamped linear elastic SDOF case, the HHT-a and Newmark BA methods are the most accurate among the advanced and the elementary methods respectively. It has to be pointed out that the total energy of the system decreases with time due to the presence of

viscous damping; therefore the reference energy curves in the subplots have negative slope and are straight because the energy decrease is exponential and its natural logarithm linear.

In Fig. 6, the natural logarithm of the total energy of the linear elastic SDOF system with damping due to Coulomb friction (benchmark problem described in section 3.5) is presented. The most accurate estimation of energy is provided by the Continuous acceleration algorithms and the Newmark Backward Acceleration method. The performance of the U1-V0-DA (<13>) is the poorest as in the previous examples. The reference energy curves are not straight lines, because the rate of energy dissipation is quadratic and not exponential, as can be seen from Eq. (31).

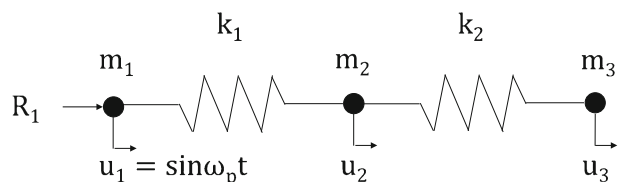
In Fig. 7 the natural logarithm of the total energy of the linear elastic SDOF system with hysteretic damping (benchmark problem described in section 3.6) is presented. Similarly to the results of the other problems, the most accurate algorithms prove to be the Continuous Acceleration algorithms. The most accurate scheme appears to be the HHT-a method among the advanced algorithms, and the Newmark Backward Acceleration scheme among the elementary algorithms. Generally, the energy plots show periodic fluctuations and the U0-V0-Opt (<8>) and U0-V0-DA (<10>) algorithms seem to perform better than their U0-V1 or U1-V0 counterparts, a general trend observed in Figs. 2, 3, 4, 5, 6 and 7. Analytical calculation of the reference total energy is not possible in this benchmark problem and for this reason it is computed by setting a much smaller time step, as outlined in Sect. 3.6. The linear configuration of the various curves implies that the energy dissipation during the free vibration of a hysteretically damped SDOF oscillator varies exponentially with time; this type of variation was previously observed also for the viscously damped SDOF oscillator presented in Fig. 5 and described in section 3.4.

It is generally observed in the numerical results in terms of energy presented above that the most accurate scheme in the linear and nonlinear regimes is the HHT-a method. The time integration algorithms considered in this study (either advanced or elementary) do not always ensure that the variation of energy is as expected (e.g. conserved for undamped and unforced systems). However, any degree of accuracy can be achieved for all types of oscillators considered in this study, if the time step is sufficiently reduced.

5 Undamped linear elastic system with 3 DOFS

In this section, the benchmark problem studied by Bathe and Noh (2012) is considered, which is a 3-degree-of-freedom spring system shown in Fig. 8. This problem is used as a benchmark problem mainly due to the fact that it represents finite element models which include high stiffness elements, as well as flexible elements. High stiffness elements can involve, for example, rigid connections or penalty factors or various types of constraints of the model. Such high stiffness values used in finite element models have rarely any physical meaning; they are almost always used to provide constraints. This system is studied here in order to compare the dynamic response produced by the various time

Fig. 8 Model problem of three degrees of freedom spring system



integration algorithms with the “exact” presented by Bathe and Noh (2012). The numerical data $k_1 = 10^7$, $k_2 = 1$, $m_1 = 0$, $m_2 = 1$, $m_3 = 1$ are used for the system properties. Also, node 1 has prescribed displacement equal to:

$$u_1 = \sin \omega_p t \quad (34)$$

where $\omega_p = 1.2$. The governing equation of motion is the following:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \\ 0 \end{bmatrix} \quad (35)$$

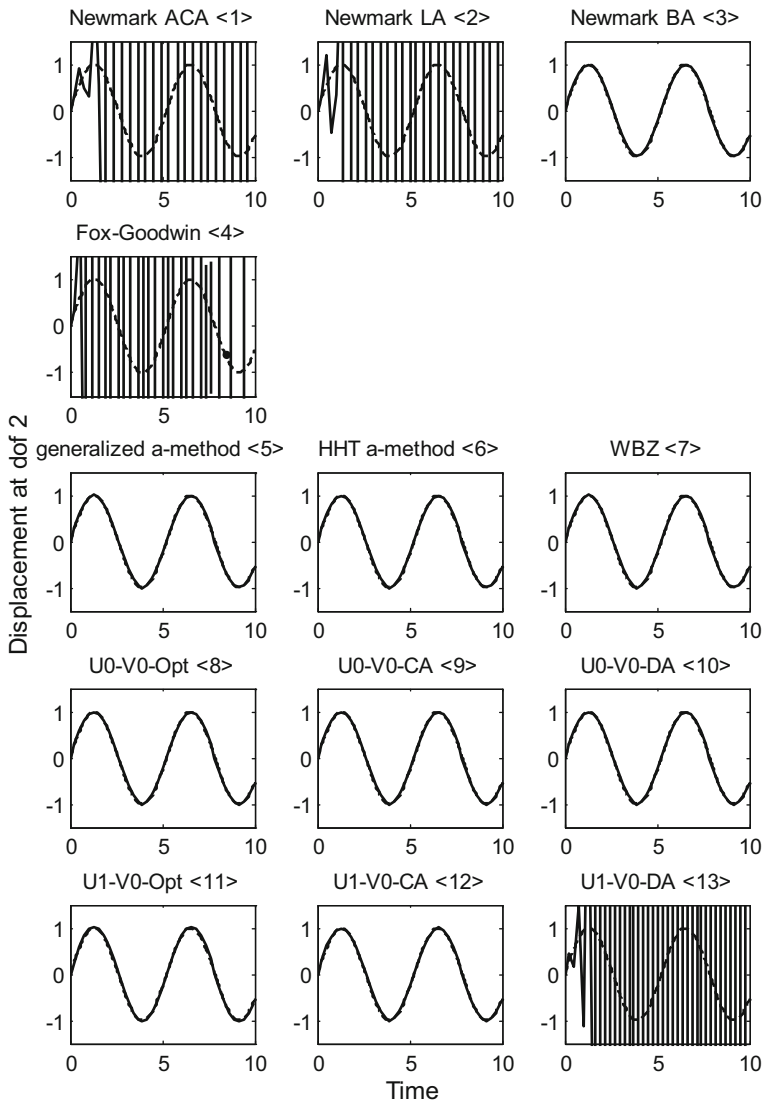


Fig. 9 Time history of the displacement at the dof 2 of the 3DOF undamped linear elastic system

which can be rewritten for the unknown displacements u_2 and u_3 as follows:

$$\begin{bmatrix} m_2 & 0 \\ 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{u}_2 \\ \ddot{u}_3 \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} k_1 u_1 \\ 0 \end{bmatrix} \quad (36)$$

The initial conditions are considered to be zero for both the displacements and the velocities. The response of the system is estimated for a time duration of 10 s, and the time step used is $\Delta t = 0.2618$ s.

The displacement, velocity and acceleration at degree of freedom 2 of the 3dof system are calculated and compared to the corresponding results of Bathe and Noh (2012). The

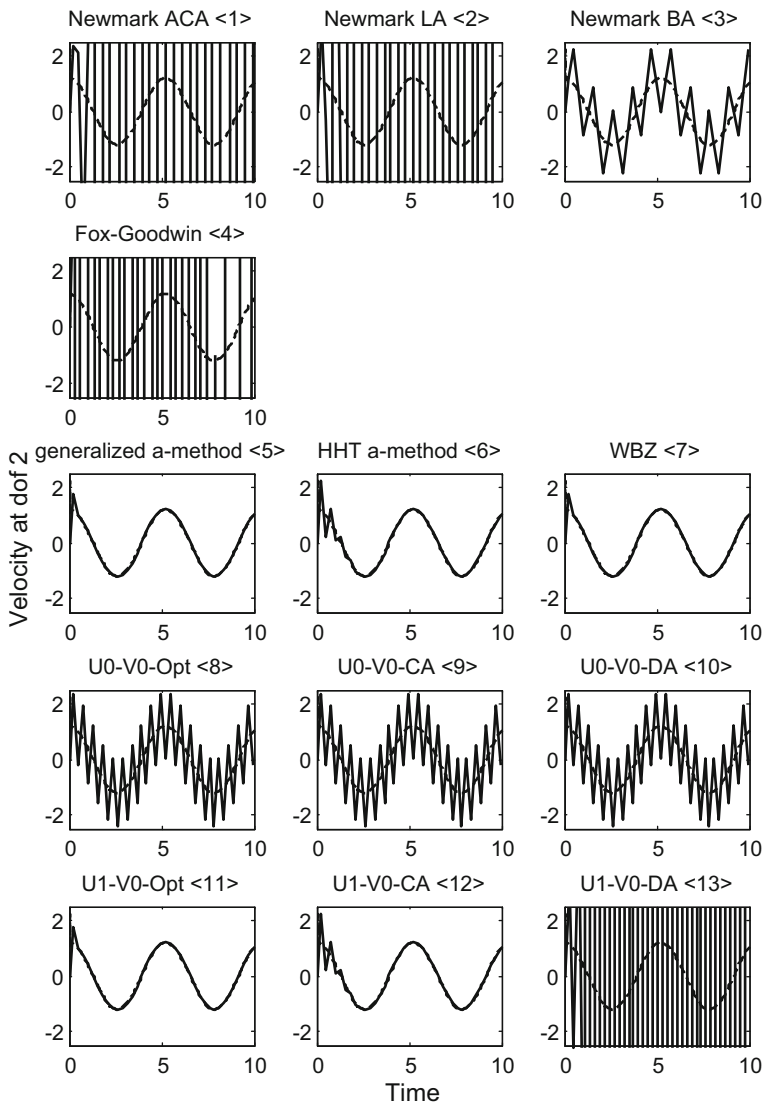


Fig. 10 Time history of the velocity at the dof 2 of the 3DOF undamped linear elastic system

displacement response of the degree of freedom 2 of the 3DOF oscillator is illustrated in Fig. 9. The exact solution, as presented by Bathe and Noh (2012) is plotted with a dashed curve. It is evident that for all advanced algorithms, except for the U1-V0-DA (<13>) algorithm, the numerically calculated displacement time history practically coincides with the exact solution. Among the elementary time integration schemes, only the Newmark Backward Acceleration method manages to trace the response, whereas all others fail. It is concluded that the majority of the advanced algorithms reproduce the exact displacement result of the time integration procedure regarding the structural system studied by Bathe and Noh (2012).

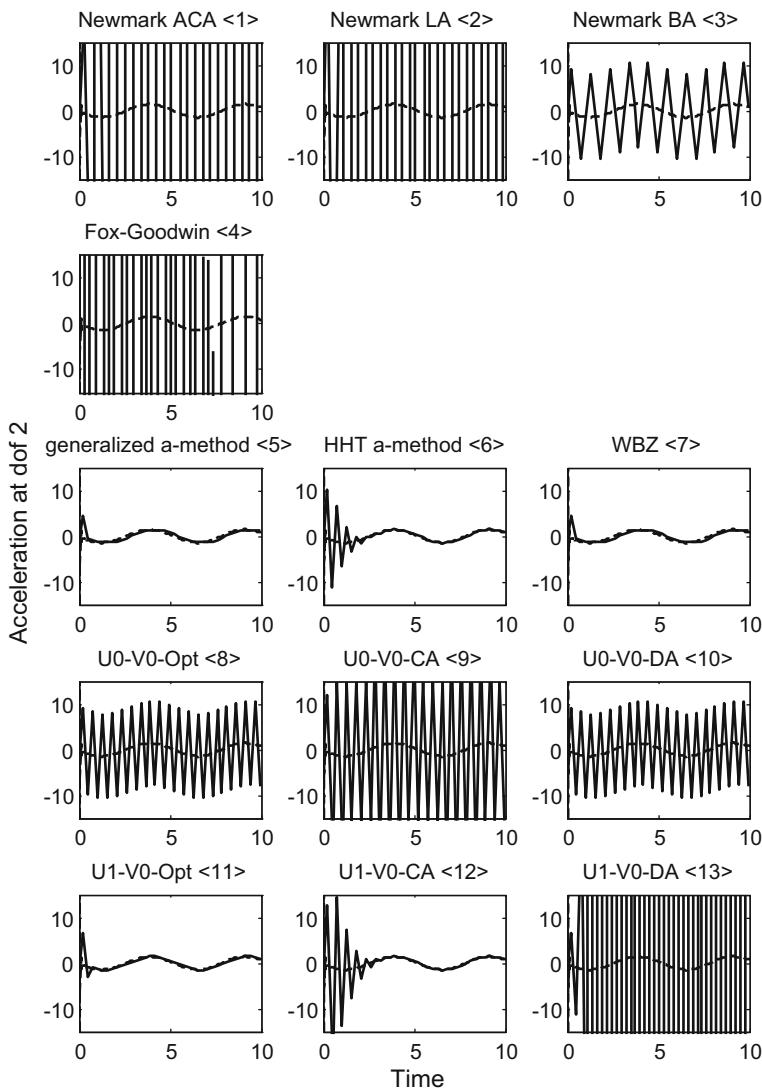


Fig. 11 Time history of the acceleration at the dof 2 of the 3DOF undamped linear elastic system

Things become different in the case of the velocity at dof 2 of the 3DOF undamped linear elastic system, as the response is computed accurately only by the U0-V1 and U1-V0 algorithms except for the U1-V0-DA (<13>) algorithm, as seen in Fig. 10. Spurious oscillations around the exact solution are observed in the response produced by the U0-V0 algorithms. All the elementary integration schemes fail to reproduce the exact velocity response; the erroneous behavior of the Newmark BA algorithm has been already shown by Bathe and Noh (2012). It seems that in spite of the fact that the HHT-a method is less accurate than the generalized- α and the WBZ methods at the beginning of the time history response, its accuracy remains comparable to that of the two other integration schemes for the later stages of the dynamic response, and therefore could be an option equally good with the generalized- α and the WBZ methods.

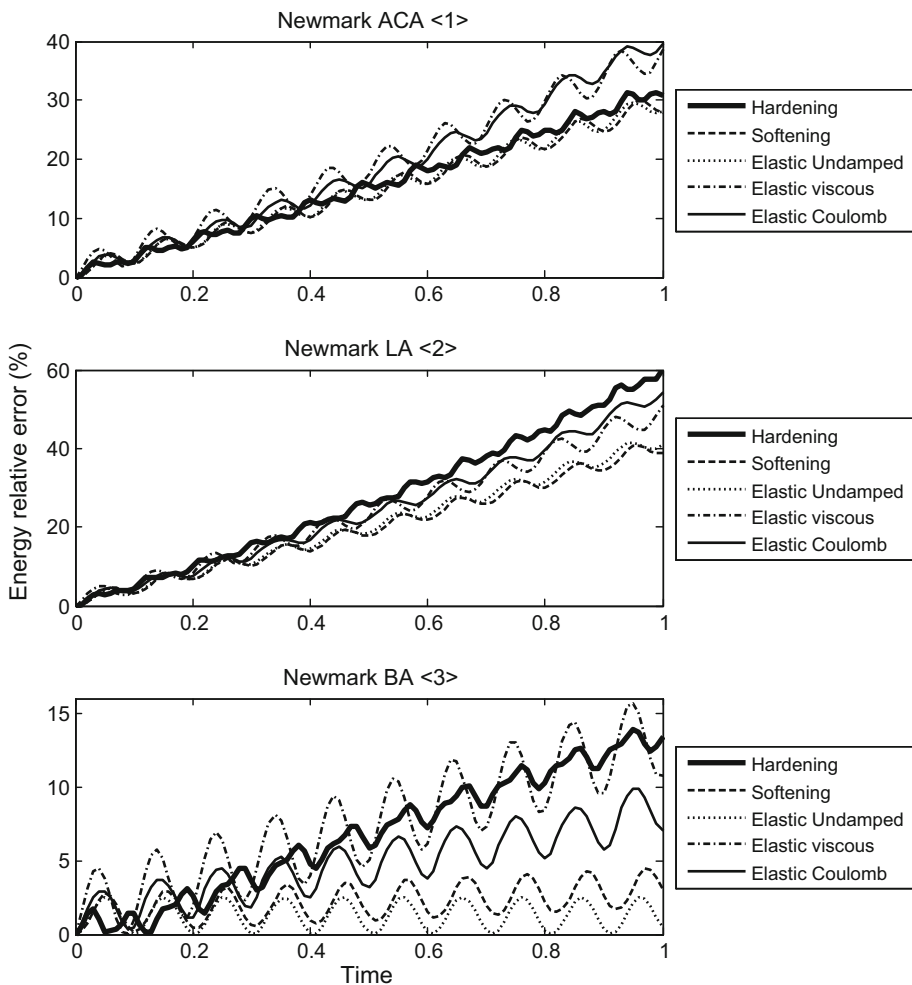


Fig. 12 Time history of the relative error of the total energy of the SDOF oscillators of problems 1–5 integrated by the Newmark ACA, Newmark LA and Newmark BA algorithms of the GSSSS family

The acceleration time history at dof 2 of the 3DOF undamped linear elastic system is shown in Fig. 11. The exact solution, as presented by Bathe and Noh (2012), is most accurately reproduced by the generalized- α , WBZ and U1-V0-Opt (<11>) methods. All the elementary time integration methods fail to accurately calculate the dynamic response. The spurious undulations which are present in certain subplots have already been pointed out by Bathe and Noh (2012).

6 Comparison of algorithm performance in linear and nonlinear regime

In this section the performance of the various time integration schemes is evaluated for linear and nonlinear problems. Comparisons are made between these two categories of analysis. Since the time step, the time duration, and the algorithmic constants are the same

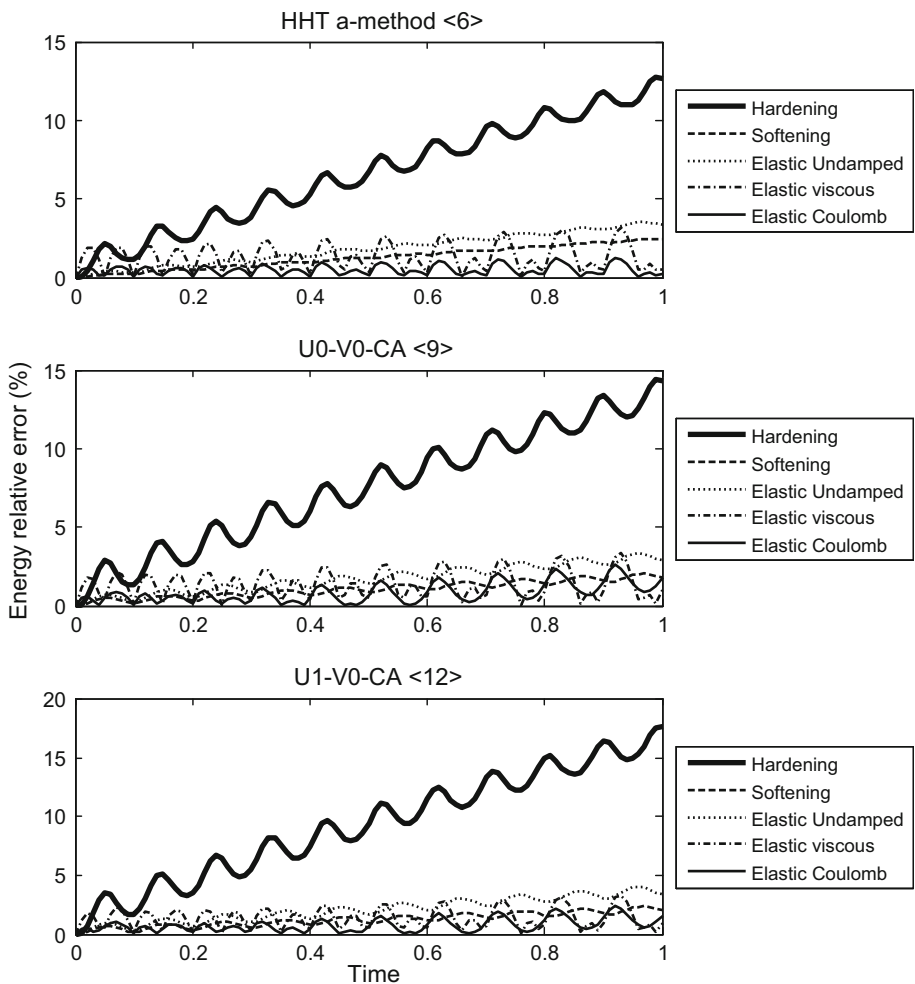


Fig. 13 Time history of the relative error of the total energy of the SDOF oscillators of problems 1–5 integrated by the continuous acceleration algorithms of the GSSSS family

for all benchmark problems analyzed, and the stiffness of the various oscillators in their small-strain region is the same, the comparisons in this section are made with reference to the quality of the time integration schemes and their associated numerical error.

In Fig. 12, a dimensionless quantity equal to the relative error of the total energy of the various systems is plotted against integration time. Three schemes are used for the integration of the SDOF systems involved: the Newmark Average Constant Acceleration (ACA), Newmark Linear Acceleration (LA) and Newmark Backward Acceleration (BA) methods. The various curves include linear as well as nonlinear energy error. It is seen that, depending on the algorithm used, the various SDOF systems yield different integration errors. In the case of Newmark ACA algorithm, the largest error is observed for the viscoelastic and elastic system with Coulomb damping, whereas the lowest energy error is observed for softening and linear elastic undamped systems. In the case of Newmark LA

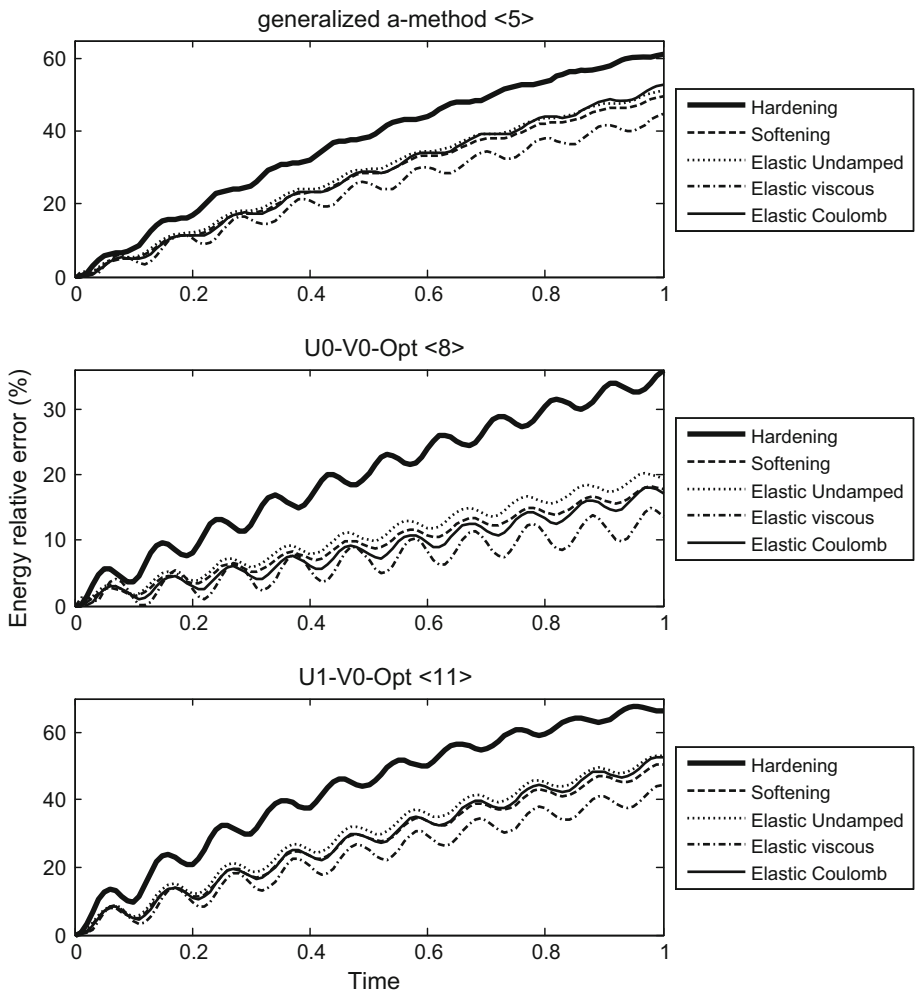


Fig. 14 Time history of the relative error of the total energy of the SDOF oscillators of problems 1–5 integrated by the optimal numerical dissipation and dispersion algorithms of the GSSS family

algorithm, the hardening system shows the maximum relative error, whereas the softening and linear elastic undamped systems show the minimum relative error. Finally, Newmark BA algorithm yields the most accurate results for linear elastic undamped system, whereas the maximum error is observed for the hardening and viscoelastic systems.

Similar to Figs. 12 and 13 compares the relative error of the total energy of the various systems, plotted against integration time, where the continuous acceleration (CA) algorithms are used. It is apparent that the hardening system yields relatively large energy numerical error for all three CA members of the GSSSS family. The maximum error is generally observed for the U1-V0-CA (<12>) algorithm, for all problems considered in this study. Compared to the error of the hardening system, the error of the remaining systems is considerably lower. The error is much lower if Continuous acceleration

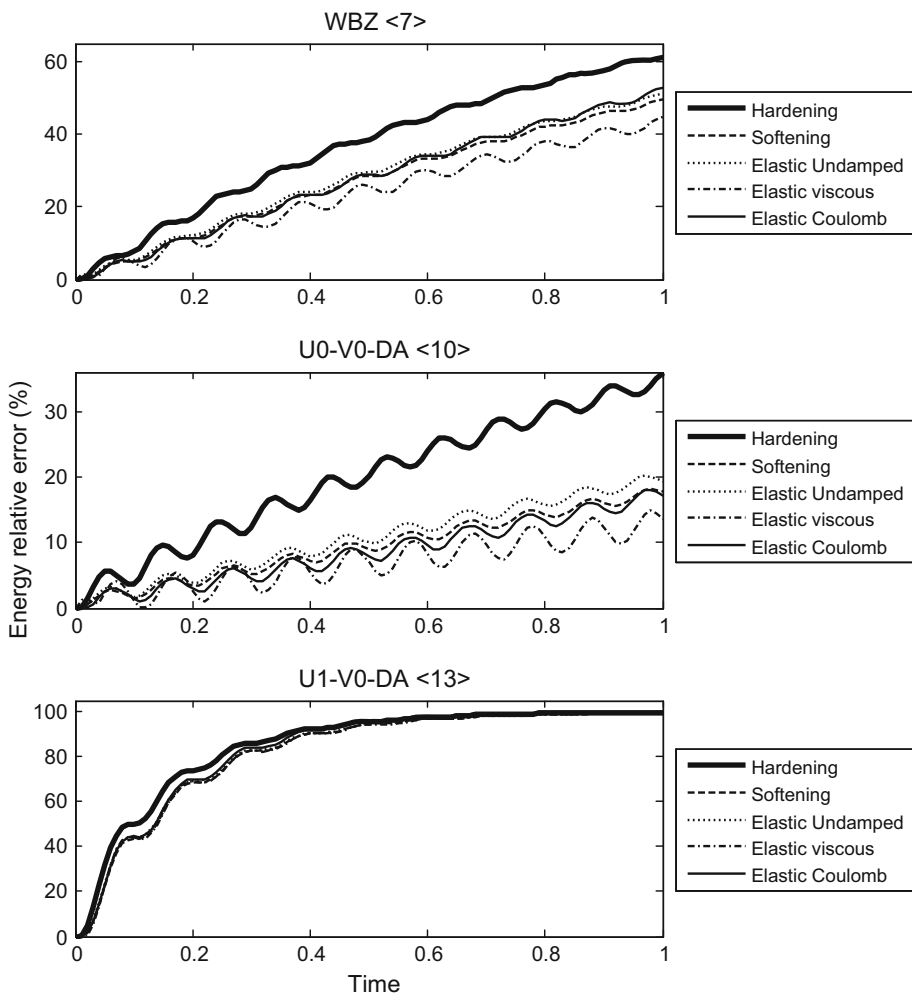


Fig. 15 Time history of the relative error of the total energy of the SDOF oscillators of problems 1–5 integrated by the discontinuous acceleration algorithms of the GSSSS family

algorithms are used, compared to the elementary time integration algorithms the results of which are shown in Fig. 12.

Corresponding results with those of the previous paragraph are observed in Figs. 14 and 15, for the optimal numerical dissipation and dispersion algorithms and the discontinuous acceleration (DA) algorithms respectively. The hardening system exhibits the maximum total energy error for all these types of algorithms, as can be seen in both figures. The lowest energy error is observed for the linear viscoelastic system for all optimum numerical dissipation and dispersion algorithms, as well as discontinuous acceleration algorithms. Finally, the error associated with the optimal numerical dissipation and dispersion algorithms and the discontinuous acceleration algorithms is certainly higher in general than that of the continuous acceleration methods, for the same SDOF models studied.

The main conclusion from this figure is that the linear and nonlinear versions of the GSSSS algorithm family yield comparable percentages of total energy error, and in many instances, the error of the total energy of nonlinear systems is lower than the error in the total energy of linear systems. Given that in each subplot the problems considered have the same initial conditions, time step, integration algorithm, nonlinear convergence parameters, etc., the differences between the various curves in each subplot originate mainly from the differences between the constitutive properties of the oscillators. It is a fact that the existence of iterations within a time step is a source of additional error. The linear integration algorithms do not require any iterations within a time step. The opposite is the case with the nonlinear time integration algorithms. Consequently, it is expected that the error accumulated in the nonlinear response will be higher than the error of the linear response. However, from the results presented in this study, it is concluded that this does not happen in a regular basis; there are cases in which the error of a nonlinear problem is lower than that of a linear problem, while the same integration scheme is used. From this observation, the conclusion is drawn that the extension of the linear versions of the GSSSS algorithms into the nonlinear regime with incorporation of Newton–Raphson iterations results in nonlinear time integration algorithms which possess a similar (and sometimes higher) level of accuracy with their linear counterparts.

The results presented in Figs. 12, 13, 14 and 15 are summarized in Fig. 16, where the maximum relative error of the total energy for each SDOF system—integration algorithm pair and for the duration of the oscillation is shown in the form of bar chart. In the vertical axis each label refers to the type of SDOF system and the time integration algorithm with which its response is calculated, whereas in the horizontal axis, the percentage of relative energy error is shown. The lowest error is observed for the Continuous Acceleration algorithms (including HHT algorithm) for the linear elastic with Coulomb damping, undamped softening, undamped linear elastic and viscoelastic oscillators and the Newmark Backward Acceleration algorithm for the undamped linear elastic oscillator. On the other hand, the error associated with the Discontinuous Acceleration algorithms and the undamped hardening oscillator is the largest among all cases and algorithms considered in this study.

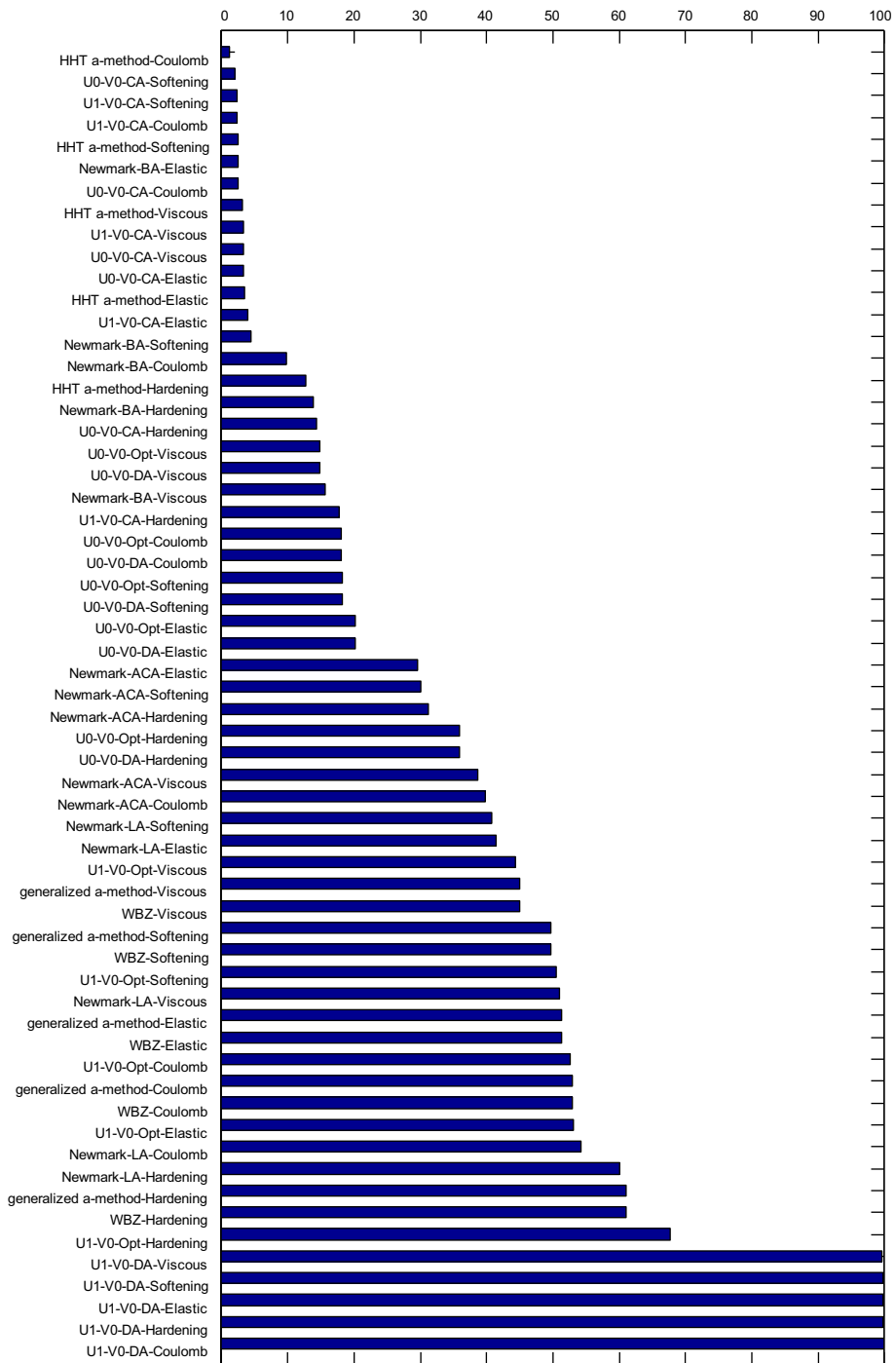


Fig. 16 Maximum relative error of the total energy of the SDOF oscillators of problems 1–5 integrated by the nonlinear integration algorithms developed in this study

7 Conclusions

The family of linear generalized single step single solve (GSSSS) algorithms, which includes the most commonly used time integration algorithms as special cases, can be extended to solve materially nonlinear dynamic response via a Newton–Raphson iterative procedure. In the nonlinear regime, the extended nonlinear generalized algorithms are very efficient, with acceptable accuracy and stability, even with increased size of the time step. They perform much better in some cases illustrated in this study, contrary to the elementary algorithms which in practice fail to trace the dynamic response.

It is pointed out that unconditionally stable algorithms for linear problems, may lose their stability in the presence of nonlinearities. For increased time step (and thus lower computational effort), the Continuous Acceleration methods, which include the HHT- α method ($\alpha < 0.5$) as a special case, exhibit the most accurate response for most of the cases studied. They appear to be the best option, regarding their general performance at the benchmark problems studied compared to the other integration schemes. For sufficiently small time step, all algorithms converge to the exact dynamic response. Further research has to be made to investigate the relation between the stable time increment of the generalized single step single solve algorithms applied in nonlinear problems and various other problem-dependent input data.

Acknowledgements This work has been supported by the European Research Council Advanced Grant MASTER—Mastering the computational challenges in numerical modeling and optimum design of CNT reinforced composites (ERC-2011-ADG-20110209).

References

- Argyris JH, Dunne PC, Angelopoulos T (1973) Dynamic response by large step integration. *Earthq Eng Struct Dyn* 2:185–203
- Armero F, Pet  cz E (1996) A new class of conserving algorithms for dynamic contact problems. In: D  s  d  ri JA, LeTallec P, O  ate E, P  riaux J, Stein E (eds) 2nd ECCOMAS conference on numerical methods in engineering, Paris, France, September 9–13
- Ascher UM, Petzold LR (1998) Computer methods for ordinary differential equations and differential-algebraic equations. SIAM, Philadelphia
- Bathe KJ (2007) Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Comput Struct* 85(7–8):437–445
- Bathe KJ, Noh G (2012) Insight into an implicit time integration scheme for structural dynamics. *Comput Struct* 98–99:1–6
- Chung J, Hulbert GM (1993) A time integration method for structural dynamics with improved numerical dissipation: the generalized α -method. *J Appl Mech* 60(2):371–375
- Clough RW, Penzien J (2003) Dynamics of structures, 3rd edn. Computers & Structures, Inc, Berkeley
- Fox L, Goodwin ET (1949) Some new methods for the numerical integration of ordinary differential equations. *Math Proc Camb Philos Soc* 45:373–388
- Hilber HM, Hughes TJR, Taylor RL (1977) Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthq Eng Struct Dyn* 5:283–292
- Hughes TJR, Caughey TK, Liu WK (1978) Finite-element methods for nonlinear elastodynamics which conserve energy. *J Appl Mech Trans ASME* 45:366–370
- Kuhl D, Crisfield MA (1999) Energy-conserving and decaying algorithms in non-linear structural dynamics. *Int J Numer Methods Eng* 45:569–599
- Kuhl D, Ramm E (1996) Constraint energy momentum algorithm and its application to nonlinear dynamics of shells. *Comput Methods Appl Mech Eng* 136:293–315
- Liu J, Wang X (2008) An assessment of the differential quadrature time integration scheme for nonlinear dynamic equations. *J Sound Vib* 314:246–253
- Newmark NM (1959) A method of computation for structural dynamics. *J Eng Mech* 85(EM3):67–94

- Rezaiee-Pajand M, Sarafrazi SR, Hashemian M (2011) Improving stability domains of the implicit higher order accuracy method. *Int J Numer Methods Eng* 88(9):880–896
- Rosales MB, Filipich CP (2002) Time integration of non-linear dynamic equations by means of a direct variational method. *J Sound Vib* 254(4):763–775
- Simo JC, Tarnow N (1992) The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *J Appl Math Phys* 43(5):757–792
- Wood WL, Bossak M, Zienkiewicz OC (1980) An alpha modification of Newmark's method. *Int J Numer Methods Eng* 15(10):1562–1566
- Xie YM (1996) An assessment of time integration schemes for non-linear dynamic equations. *J Sound Vib* 192(1):321–331
- Zhang L, Liu T, Li Q (2014) A robust and efficient composite time integration algorithm for nonlinear structural dynamic analysis. *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, Cairo
- Zhou X, Tamma KK (2004) Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *Int J Numer Methods Eng* 59:597–668
- Zhu F, Wang J-T, Jin F, Gui Y (2016) Comparison of explicit integration algorithms for real-time hybrid simulation. *Bull Earthq Eng* 14(1):89–114