COMPDYN 2015
5th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, V. Papadopoulos, V. Plevris (eds.)
Crete Island, Greece, 25–27 May 2015

NONLINEAR DYNAMIC RESPONSE OF HARDENING, SOFTENING AND ELASTOPLASTIC SDOF SYSTEMS USING GENERALIZED SINGLE STEP ALGORITHMS WITH NEWTON – RAPHSON ITERATIONS

George Papazafeiropoulos[1], Vagelis Plevris[2], and Manolis Papadrakakis[1]

[1] National Technical University of Athens
Institute of Structural Analysis and Antiseismic Research
Zografou Campus, Athens 15780, Greece
gpapazafeiropoulos@yahoo.gr, mpapadra@central.ntua.gr

[2] Computational Mechanics Laboratory, Department of Civil Engineering
School of Pedagogical & Technological Education
Heraklion, Athens, 141 21, Greece
vplevris@aspete.gr

Keywords: Nonlinear dynamic response, Newton-Raphson, stability, elastic, plastic, hardening, softening, elastoplastic, single step time integration algorithm.

Abstract. *The dynamic equation of motion of a SDOF system with a nonlinear elastic hardening spring, a SDOF system with a nonlinear elastic softening spring and a SDOF system with a nonlinear elastic-plastic spring is integrated numerically using a family of linear generalized single step-single solve algorithms. For this purpose, these algorithms are extended by using a Newton-Raphson type iterative procedure in each time step to ensure dynamic equilibrium. After a literature review of the available time integration schemes used for nonlinear problems, the linear family of algorithms is presented along with several common time integration algorithms as special cases of the generalized algorithm. An explicit flowchart is given showing the integration procedure used in the present study. The modified algorithm is applied to the aforementioned three types of SDOF systems and results concerning phase portraits, (relative) energy decrease, iterations needed for equilibrium and internal force - displacement curves are presented. It is shown that the algorithms with optimal numerical dissipation and dispersion perform in general better than others, and that from the algorithms with optimal numerical dissipation and dispersion, only the one with zero-displacement and zero-velocity overshooting behavior can capture efficiently the elastoplastic dynamic response.*

1 INTRODUCTION

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

$$M\ddot{u} + p\left(u,\dot{u}\right) = f\left(t\right) \tag{1}$$

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

$$p\left(u,\dot{u}\right) = \bar{K}u + \bar{C}\dot{u} \tag{2}$$

where $\bar{K}$ is the stiffness matrix and $\bar{C}$ 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 $p\left(u,\dot{u}\right)$ 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 step-by-step methods.

Direct time integration (or time stepping, or step by step) methods are a widely used approach to solve dynamic linear or nonlinear response analysis problems. In these methods the equilibrium relations 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 are the following:

- Stability. An integration scheme is said to be stable if the numerical solution, under any initial conditions, does not grow without bound. An algorithm is unconditionally stable for linear problems if the convergence of the solution is independent of the size of the time step.
- Convergence. An integration scheme is convergent if the numerical solution approaches the exact solution as the size of the time step tends to zero.
- Accuracy. Two numerical errors are associated with the accuracy of any algorithm: (a) numerical dispersion (often expressed in terms of period elongation) and (b) numerical dissipation (often expressed in terms of either the amplitude decay or the algorithmic damping ratio).
- Algorithmic dissipation. It is a kind of filtering of the higher frequency oscillations, necessary to eliminate the spurious high frequency modes inherent in a finite element mesh.
- Self-starting. This type of algorithms requires data from two time steps to proceed the solution. If data from more than two time steps are needed, the algorithm must be implemented with a starting procedure.
- Overshooting. It is the tendency of an algorithm to exceed heavily the exact solution in the first few time steps, but eventually to converge to the exact solution.

Taking into account the above characteristics, a time integration scheme should have the following desirable features [1]:

- Unconditional stability.
- At least second-order accuracy in time.
- No more than zero-order displacement and velocity overshooting behavior with minimal numerical dissipation and dispersion.
- Self-starting features with no more than one set of single-field system of implicit equations to be solved at each time step to include ease of implementation and computational simplicity.

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 such cases.

In this study, after a concise literature review about the numerical direct time integration algorithms applicable to the dynamic equilibrium equations of structural analysis of the form (1), a general single step linear integration group of algorithms is presented, which incorporates many well-known algorithms, and which is modified to account for nonlinear response, by introducing a Newton-Raphson iterative procedure at each time step. Afterwards, the modified algorithm is applied to known nonlinear dynamic analysis example problems and the results are studied. Apart from its robustness in solving nonlinear problems, it is proved that the algorithm can be designed to cope with cases with any degree of nonlinearity.

2 LITERATURE REVIEW

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 [2]. 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 the latter so that the introduced loading history fits best the actual one.

The numerical direct time integration methods 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 at each time step have improved properties [3], but they require twice or more the computational effort of simpler methods.

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 [4]. This energy criterion is summarized in the following inequality:

$$\left(U_{n+1}+K_{n+1}\right)-\left(U_{n}+K_{n}\right)\leq W_{ext} \tag{3}$$

where U_{n+1} and U_n represent the strain energy at the end and at the beginning of the time step respectively, K_{n+1} and K_n 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.
- Algorithms extending others by using constraints of energy conservation imposed via Lagrange multipliers (Constraint Energy Method), the first of which was presented in [5].
- Algorithms which enforce energy conservation algorithmically such as the energy-momentum method presented in [6]. In the absence of external loading these algorithms are designed to obey the following laws:

$$\frac{dL_t}{dt} = 0, \qquad \frac{dJ_t}{dt} = 0, \qquad \frac{dE_t^{tot}}{dt} \le 0 \tag{4}$$

where L_t is the linear momentum, 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. The Constraint Energy Momentum Algorithm developed in [7] combines numerical dissipation algorithms and enforced energy conservation algorithms, whereas combinations of numerical dissipation algorithms and algorithms which ensure energy conservation algorithmically are presented in [4,11].

Another alternative for more accurate and stable time integration algorithms is the concept of composition, namely the combination of two or more algorithms which gives other composite algorithms which are more efficient. 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 in [8], whereas a classification of these methods is presented in [9].

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. This method is outlined in [10].

3 MODIFIED NONLINEAR TIME INTEGRATION ALGORITHM

3.1 The linear generalized single step single solve algorithm

The equation of motion of a Single Degree of Freedom (SDOF) linear structure is given by the combination of the SDOF counterparts of (1) and (2):

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = f(t) \tag{5}$$

with initial conditions:

$$u(0) = u_0, \quad \dot{u}(0) = \dot{u}_0 \tag{6}$$

Equation (5) can be applied to MDOF structures, given that the latter can be decomposed into a finite number of SDOF structures using various superposition methods. In [1] 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:

$$\int_0^{\Delta t} W^T \left(M\ddot{u} + C\dot{u} + Ku - f \right) dt \tag{7}$$

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

$$W = 1 + w_1\Gamma + w_2\Gamma^2 + w_3\Gamma^3 \tag{8}$$

and:

$$\Gamma = \overline{\tau} / \Delta t, \quad \overline{\tau} = t - t_n, \quad \Delta t = t_{n+1} - t_n \tag{9}$$

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

$$u = u_n + \Lambda_1 \dot{u}_n \tau + \Lambda_2 \ddot{u}_n \tau^2 + \Lambda_3 \frac{\ddot{u}_{n+1} - \ddot{u}_n}{\Delta t} \tau^3 \tag{10}$$

$$\dot{u} = \dot{u}_n + \Lambda_4 \ddot{u}_n \tau + \Lambda_5 \frac{\ddot{u}_{n+1} - \ddot{u}_n}{\Delta t} \tau^2 \tag{11}$$

$$\ddot{u} = \ddot{u}_n + \Lambda_6 \frac{\ddot{u}_{n+1} - \ddot{u}_n}{\Delta t} \tau \tag{12}$$

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

$$f = f_n + \frac{f_{n+1} - f_n}{\Delta t} \tau \tag{13}$$

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

$$u_{n+1} = u_n + \lambda_1 \dot{u}_n \Delta t + \lambda_2 \ddot{u}_n \Delta t^2 + \lambda_3 \left(\ddot{u}_{n+1} - \ddot{u}_n \right) \Delta t^2 \tag{14}$$

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

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

$$\begin{aligned} &\left(W_1\Lambda_6 M + W_2\Lambda_5 C\Delta t + W_3\Lambda_3 K\Delta t^2 \right) \ddot{u}_{n+1} = \\ &-M\left(\ddot{u}_n - W_1\Lambda_6 \ddot{u}_n \right) \\ &-C\left(\dot{u}_n + W_1\Lambda_4 \ddot{u}_n \Delta t - W_2\Lambda_5 \ddot{u}_n \Delta t \right) \\ &-K\left(u_n + W_1\Lambda_1 \dot{u}_n \Delta t + W_2\Lambda_2 \ddot{u}_n \Delta t^2 - W_3\Lambda_3 \ddot{u}_n \Delta t^2 \right) \\ &+\left(1 - W_1 \right) f_n + W_1 f_{n+1} \end{aligned} \tag{16}$$

where the constants W_i are given by:

$$W_i = \frac{\sum_{j=0}^{3} \frac{w_j}{1+i+j}}{\sum_{j=0}^{3} \frac{w_j}{1+j}}, \quad i = 1,2,3 \tag{17}$$

There are 12 independent integration constants that are needed in order to apply the equations (16), (14) and (15) to proceed to the next step: W_1, $W_1\Lambda_1$, $W_2\Lambda_2$, $W_3\Lambda_3$, $W_1\Lambda_4$, $W_2\Lambda_5$, $W_1\Lambda_6$, λ_1, λ_2, λ_3, λ_4, λ_5. Each set of these parameters defines the algorithm uniquely, 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. In [1], the integration parameters are calculated by imposing several different constraints to the algorithm, regarding order of accuracy, overshooting behavior (in terms of displacement and velocity orders), spurious roots at the high and low frequency limits, dissipation and dispersion properties, 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 (17), after specifying the parameters w_1, w_2, w_3.

3.2 Design of the linear generalized single step single solve algorithm – special cases

An algorithm is termed to have the property of continuous acceleration, if the acceleration $\ddot{u}_{n+1}$ 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 [1].

The procedure for designing the algorithm presented in the previous section to apply it to time integration problems (i.e. setting its 12 integration constants), is presented in [1]. The algorithms of the generalized single step family are the following:

- Zero-order displacement, zero-order velocity, optimal numerical dissipation and dispersion (U0-V0-Opt)
- Zero-order displacement, zero-order velocity, continuous acceleration (U0-V0-CA)
- Zero-order displacement, zero-order velocity, discontinuous acceleration (U0-V0-DA)
- Zero-order displacement, first-order velocity, optimal numerical dissipation and dispersion (U0-V1-Opt)
- Zero-order displacement, first-order velocity, continuous acceleration (U0-V1-CA)
- Zero-order displacement, first-order velocity, discontinuous acceleration (U0-V1-DA)
- First-order displacement, zero-order velocity, optimal numerical dissipation and dispersion (U1-V0-Opt)
- First-order displacement, zero-order velocity, continuous acceleration (U1-V0-CA)
- First-order displacement, zero-order velocity, discontinuous acceleration (U1-V0-DA)

The values of the parameters are shown for various well-known integration schemes in tables. In Table 1 the parameters of the central difference method, the average constant acceleration method [12], the linear acceleration method [12], the Fox-Goodwin formula [14], the backward acceleration method [13] and the general family of Newmark methods [12]. In Table 2 the parameters of the zero-order displacement and zero-order velocity overshooting algorithms are presented [1]. In order to evaluate the integration constants, the quantity $\rho_{\inf}$, 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 appropriate row of the table. If $\rho_{\inf}=1$, the resulting algorithm is non-dissipative. In Table 3 the parameters are given for the zero-order displacement, first-order velocity overshooting algorithms, presented in [1]. 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 a-method, the HHT-a method and the WBZ a-method, presented in [15,16,17] respectively. Table 4 shows the parameters of the first-order displacement and zero-order velocity overshooting algorithms. If $\rho_{\inf}=1$, the first-order displacement, zero-order velocity, optimal numerical dissipation and dispersion, the first-

	Central Difference Method	Average constant acceleration [12]	Linear Acceleration Method [12]	Fox-Goodwin formula [14]	Backward acceleration method [13]	Family of Newmark Methods [12]
w_1	-15	-15	-15	-15	-15	-15
w_2	45	45	45	45	45	45
w_3	-35	-35	-35	-35	-35	-35
$W_1\Lambda_1$	1	1	1	1	1	1
$W_2\Lambda_2$	0	0.25	1/6	1/12	0.5	β
$W_3\Lambda_3$	0	0.25	1/6	1/12	0.5	β
$W_1\Lambda_4$	0.5	0.5	0.5	0.5	0.5	γ
$W_2\Lambda_5$	0.5	0.5	0.5	0.5	0.5	γ
$W_1\Lambda_6$	1	1	1	1	1	1
λ_1	1	1	1	1	1	1
λ_2	0.5	0.5	0.5	0.5	0.5	0.5
λ_3	0	0.25	1/6	1/12	0.5	β
λ_4	1	1	1	1	1	1
λ_5	0.5	0.5	0.5	0.5	0.5	γ

Table 1: Integration parameters for various known time integration schemes.

	U0-V0-Opt	U0-V0-CA	U0-V0-DA
ρinf	[0 1]	[1/3 1]	[0 1]
w_1	-15(1-2ρinf)/(1-4ρinf)	-15(1-5ρinf)/(3-7ρinf)	-15
w_2	15(3-4ρinf)/(1-4ρinf)	15(1-13ρinf)/(3-7ρinf)	45
w_3	-35(1-ρinf)/(1-4ρinf)	140ρinf/(3-7ρinf)	-35
$W_1\Lambda_1$	1/(1+ρinf)	(1+3ρinf)/2/(1+ρinf)	1
$W_2\Lambda_2$	1/2/(1+ρinf)	(1+3ρinf)/4/(1+ρinf)	1/2
$W_3\Lambda_3$	1/2/(1+ρinf)^2	(1+3ρinf)/4/(1+ρinf)^2	1/2/(1+ρinf)
$W_1\Lambda_4$	1/(1+ρinf)	(1+3ρinf)/2/(1+ρinf)	1
$W_2\Lambda_5$	1/(1+ρinf)^2	(1+3ρinf)/2/(1+ρinf)^2	1/(1+ρinf)
$W_1\Lambda_6$	(3-ρinf)/2/(1+ρinf)	1	(3+ρinf)/2/(1+ρinf)
λ_1	1	1	1
λ_2	1/2	1/2	1/2
λ_3	1/2/(1+ρinf)	1/2/(1+ρinf)	1/2/(1+ρinf)
λ_4	1	1	1
λ_5	1/(1+ρinf);	1/(1+ρinf)	1/(1+ρinf)

Table 2: Integration parameters for zero-order displacement, zero order velocity overshooting algorithms.

	U0-V1-Opt [15]	U0-V1-CA [16]	U0-V1-DA [17]
ρinf	[0 1]	[1/2 1]	[0 1]
w_1	-15(1-2ρinf)/(1-4ρinf)	-15(1-2ρinf)/(2-3ρinf)	-15
w_2	15(3-4ρinf)/(1-4ρinf)	15(2-5ρinf)/(2-3ρinf)	45
w_3	-35(1-ρinf)/(1-4ρinf)	-35(1-3ρinf)/2/(2-3ρinf)	-35
$W_1\Lambda_1$	1/(1+ρinf)	2ρinf/(1+ρinf)	1
$W_2\Lambda_2$	1/2/(1+ρinf)	ρinf/(1+ρinf)	1/2
$W_3\Lambda_3$	1/(1+ρinf)^3	2ρinf/(1+ρinf)^3	1/(1+ρinf)^2
$W_1\Lambda_4$	1/(1+ρinf)	2ρinf/(1+ρinf)	1
$W_2\Lambda_5$	(3-ρinf)/2/(1+ρinf)^2	ρinf(3-ρinf)/(1+ρinf)^2	(3-ρinf)/2/(1+ρinf)
$W_1\Lambda_6$	(2-ρinf)/(1+ρinf)	1	2/(1+ρinf)
λ_1	1	1	1
λ_2	1/2	1/2	1/2
λ_3	1/(1+ρinf)^2	1/(1+ρinf)^2	1/(1+ρinf)^2
λ_4	1	1	1
λ_5	(3-ρinf)/2/(1+ρinf)	(3-ρinf)/2/(1+ρinf)	(3-ρinf)/2/(1+ρinf)

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

	U1-V0-Opt	U1-V0-CA	U1-V0-DA
ρinf	[0 1]	[1/2 1]	[0 1]
w_1	-30(3-8ρinf+6ρinf^2) / (9-22ρinf+19ρinf^2)	-60(2-8ρinf+7ρinf^2) / (11-48ρinf+41ρinf^2)	-30(3-4ρinf)/(9-11ρinf)
w_2	15(25-74ρinf+53ρinf^2) / 2 / (9-22ρinf+19ρinf^2)	15(37-140ρinf+127ρinf^2) / 2 / (11-48ρinf+41ρinf^2)	15(25-37ρinf)/2/(9-11ρinf)
w_3	-35(3-10ρinf+7ρinf^2) / (9-22ρinf+19ρinf^2)	-35(5-18ρinf+17ρinf^2) / (11-48ρinf+41ρinf^2)	-35(3-5ρinf) / (9-11ρinf)
$W_1\Lambda_1$	(3-ρinf)/2/(1+ρinf)	(1+3ρinf)/2/(1+ρinf)	(3+ρinf)/2/(1+ρinf)
$W_2\Lambda_2$	1/(1+ρinf)^2	2ρinf/(1+ρinf)^2	1/(1+ρinf)
$W_3\Lambda_3$	1/(1+ρinf)^3	2ρinf/(1+ρinf)^3	1/(1+ρinf)^2
$W_1\Lambda_4$	(3-ρinf)/2/(1+ρinf)	(1+3ρinf)/2/(1+ρinf)	(3+ρinf)/2/(1+ρinf)
$W_2\Lambda_5$	2/(1+ρinf)^3	4ρinf/(1+ρinf)^3	2/(1+ρinf)^2
$W_1\Lambda_6$	(2-ρinf)/(1+ρinf)	1	2/(1+ρinf)
λ_1	1	1	1
λ_2	1/2	1/2	1/2
λ_3	1/2/(1+ρinf)	1/2/(1+ρinf)	1/(1+ρinf)^2
λ_4	1	1	1
λ_5	1/(1+ρinf)	1/(1+ρinf)	(3-ρinf)/2/(1+ρinf)

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

order displacement, zero-order velocity, continuous acceleration and the first-order displacement, zero-order velocity, discontinuous acceleration algorithms recover, the first the midpoint rule a-form algorithm, and the two last the Newmark average acceleration a-form algorithm.

3.3 Modification of the linear algorithm for nonlinear dynamic response

In this section the generalized linear family of algorithms presented above is modified to account for materially nonlinear dynamic response. To proceed from the current step $(u_n, \dot{u}_n, \ddot{u}_n)$ to the next time step $(u_{n+1}, \dot{u}_{n+1}, \ddot{u}_{n+1})$, the secant stiffness and damping matrices are needed, which in general depend on u_{n+1} and $\dot{u}_{n+1}$. 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 Figure 1. The given data are the mass, stiffness and damping properties of the SDOF oscillator and the imposed external force, denoted by $M, \bar{K}, \bar{C}, f$ respectively.

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

4 BENCHMARK PROBLEMS STUDIED

4.1 Comparison of the different time integration schemes used

In this section, 13 different time integration schemes presented in the Tables 1-4 are compared through their application for solving a number of benchmark problems. The schemes compared are the average constant acceleration method [12], the linear acceleration method [12], the Fox-Goodwin formula [14], the backward acceleration method [13], the U0-V0-Opt, the U0-V0-CA, the U0-V0-DA, the U0-V1-Opt, the U0-V1-CA, the U0-V1-DA, the U1-V0-Opt, the U1-V0-CA, and the U1-V0-DA algorithms. The last 9 integration schemes are presented in [1] and details about their notation can be seen in Tables 2-4. Two of the benchmark problems involve the time integration of the equation of motion of nonlinear SDOF dynamic systems which are unforced and undamped (no external excitation is applied to them and no damping forces exist during their oscillation respectively), and at the third benchmark problem a SDOF oscillator is harmonically excited by imposing an external force with a sinusoidal time history. The response of the SDOF spring in the last case is elastoplastic, with a linear elastic and a perfectly plastic branches, exhibiting an identical yield limit in both tension and compression. An efficient nonlinear time integration scheme should conserve total energy in the first two cases, since in the last case the energy is dissipated through hysteretic response of the oscillator spring (its force-displacement diagram forms a closed loop). Thus an appropriate measure of the inaccuracy of the various time integration schemes employed can be the deviation of the total energy from its initial value:

$$e = \left| \frac{E - E_0}{E_0} \right| \tag{18}$$

in which E and E_0 are the current and initial total energy, respectively.

Three benchmark problems are solved with the various integration schemes developed in this study: the SDOF oscillator with hardening spring, the SDOF oscillator with softening spring and the elastoplastic oscillator. The two first of these examples were also studied in [18] for an assessment of the performance of various time integration schemes and in [19] for testing the differential quadrature time integration scheme, which performed successfully.

Initialize $u_n = u_0, \dot{u}_n = \dot{u}_0$

Find $K_0 = \bar{K}\left(u_0, \dot{u}_0\right)$, $C_0 = \bar{C}\left(u_0, \dot{u}_0\right)$ and $p_0 = p\left(u_0, \dot{u}_0\right)$ from (2)

Find $\ddot{u}_0 = \left(f_0 - p_0\right)/M$ 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 $n = 1 : length(f) - 1$

Initialize iteration counter $k = 1$

Initialize convergence tolerance $tol = tol_{\max}$

Initial estimate of $\ddot{u}_{n+1}^1$ at next time step ($n+1$) and first iteration ($k = 1$), from (16)

Update u_{n+1}^1 and $\dot{u}_{n+1}^1$ according to (14) and (15) respectively.

Find $K_{n+1}^1 = \bar{K}\left(u_{n+1}^1, \dot{u}_{n+1}^1\right)$, $C_{n+1}^1 = \bar{C}\left(u_{n+1}^1, \dot{u}_{n+1}^1\right)$ and $p_{n+1}^1 = p\left(u_{n+1}^1, \dot{u}_{n+1}^1\right)$ from (2)

while $tol \geq tol_{\max}$ & $k \leq k_{\max}$

Set $K_{n+1}^{k+1} = K_{n+1}^k$, $C_{n+1}^{k+1} = C_{n+1}^k$, $p_{n+1}^{k+1} = p_{n+1}^k$, $u_{n+1}^{k+1} = u_{n+1}^k$, $\dot{u}_{n+1}^{k+1} = \dot{u}_{n+1}^k$, for the next iteration

Find $\ddot{u}_{n+1}^{k+1}$ from (16)

Find the residual $R_{n+1}^{k+1} = \ddot{u}_{n+1}^{k+1} - \ddot{u}_{n+1}^1$

Set da equal to a very small quantity (infinitesimal variation of acceleration)

Find $d\ddot{u}_{n+1}^{k+1} = \ddot{u}_{n+1}^{k+1} + da$

Find $du_{n+1}^{k+1}, d\dot{u}_{n+1}^{k+1}$ from (14) and (15) respectively, using $u_{n+1}^1, \dot{u}_{n+1}^1, d\ddot{u}_{n+1}^{k+1}$

Find $dK_{n+1}^{k+1} = \bar{K}\left(du_{n+1}^{k+1}, d\dot{u}_{n+1}^{k+1}\right)$, $dC_{n+1}^{k+1} = \bar{C}\left(du_{n+1}^{k+1}, d\dot{u}_{n+1}^{k+1}\right)$ and $dp_{n+1}^{k+1} = p\left(du_{n+1}^{k+1}, d\dot{u}_{n+1}^{k+1}\right)$ from (2)

Update $d\ddot{u}_{n+1}^{k+1}$ according to (16)

Find the residual $dR_{n+1}^{k+1} = d\ddot{u}_{n+1}^{k+1} - \ddot{u}_{n+1}^1$

Find $tol = \dfrac{da}{\ddot{u}_{n+1}^1 - \ddot{u}_n} \Big/ \dfrac{dR_{n+1}^{k+1} - R_{n+1}^{k+1}}{R_{n+1}^{k+1}}$

Update $\ddot{u}_{n+1}^{k+1} = \ddot{u}_{n+1}^{k+1} - tol\left(\ddot{u}_{n+1}^{k+1} - \ddot{u}_n\right)$

Update u_{n+1}^{k+1} and $\dot{u}_{n+1}^{k+1}$ according to (14) and (15) respectively

Find $K_{n+1}^{k+1} = \bar{K}\left(u_{n+1}^{k+1}, \dot{u}_{n+1}^{k+1}\right)$, $C_{n+1}^{k+1} = \bar{C}\left(u_{n+1}^{k+1}, \dot{u}_{n+1}^{k+1}\right)$ and $p_{n+1}^{k+1} = p\left(u_{n+1}^{k+1}, \dot{u}_{n+1}^{k+1}\right)$ from (2)

Update iteration counter $k = k + 1$

end

Update $K_n = K_{n+1}^{k+1}$, $C_n = C_{n+1}^{k+1}$, $p_n = p_{n+1}^{k+1}$, $u_n = u_{n+1}^{k+1}$, $\dot{u}_n = \dot{u}_{n+1}^{k+1}$, $\ddot{u}_n = \ddot{u}_{n+1}^{k+1}$

end

Figure 1: Flowchart of the modified algorithm used in this study.

4.2 SDOF oscillator with hardening spring

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

$$\ddot{u} + S_1 u\left(1 + S_2 u^2\right) = 0 \tag{19}$$

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 (19):

$$E = \frac{1}{2}\dot{u}^2 + \frac{1}{2}S_1 u^2 + \frac{1}{4}S_1 S_2 u^4 \tag{20}$$

In this example S_1=100, S_2=10 and the initial conditions are $u_0 = 1.5$ and $\dot{u}_0 = 0$. The time step used is Δt=0.005 and the duration of the dynamic analysis is equal to 200 time steps. The $\rho_{\inf}$ parameter is selected to be equal to unity for all integration algorithms used. Concerning the Newton-Raphson iterative procedure used, the maximum convergence tolerance and the maximum number of iterations are $tol_{\max} = 0.01$ and $k_{\max} = 200$ respectively.

4.3 SDOF oscillator with softening spring

The second benchmark problem studied is the unforced and undamped SDOF oscillator with softening spring, for which the nonlinear dynamic equation of motion is:

$$\ddot{u} + S\tanh\left(u\right) = 0 \tag{21}$$

The system is conservative and its total energy is constant and given by analytical integration of (21):

$$E = \frac{1}{2}\dot{u}^2 + S\ln\left(\cosh\left(u\right)\right) \tag{22}$$

In this example S=100 and the initial conditions are $u_0 = 4$ and $\dot{u}_0 = 0$. The time step used is Δt=0.05 and the duration of the dynamic analysis is equal to 200 time steps. The $\rho_{\inf}$ parameter is selected to be equal to unity for all integration algorithms used. Concerning the Newton-Raphson iterative procedure used, the maximum convergence tolerance and the maximum number of iterations are $tol_{\max} = 0.01$ and $k_{\max} = 200$ respectively.

4.4 SDOF oscillator with elastoplastic spring

The third benchmark problem studied is the harmonically forced SDOF oscillator with elastoplastic spring, for which the initial conditions are $u_0 = 0$ and $\dot{u}_0 = 0$. The time step used is Δt=0.005 and the duration of the dynamic analysis is equal to 800 time steps. The $\rho_{\inf}$ parameter is selected to be equal to unity for all integration algorithms used. Concerning the Newton-Raphson iterative procedure used, the maximum convergence tolerance and the maximum number of iterations are $tol_{\max} = 0.01$ and $k_{\max} = 200$ respectively. As regards the elastoplastic behavior of the spring, the yield limit is assumed equal to 100 for both tension and compression and the modulus of elasticity in the linear elastic range is assumed to be equal to 50.

5 TIME INTEGRATION RESULTS

5.1 Phase portraits

The time integration results are presented in terms of phase portraits, which provide an overview of the evolution of the total energy of the oscillator. In each phase portrait, the oscillation begins from the rightmost border of each diagram, and rotates clockwise around the maxima and minima of the displacement and velocity, thus inscribing an ellipse. Maximum kinetic energy in each cycle occurs at the upper and lower extrema, whereas maximum potential occurs at the left and right extrema. If the total energy of the oscillator remains constant, the ellipse is supposed to have a single line along its thickness. In the cases studied here, the total energy decreases gradually as the oscillation proceeds, resulting in decreasing kinetic and potential energy maxima and minima (in terms of absolute values), thus leading to ellipses of lower area being inscribed into the initial one. The main reason for the energy decrease is that the convergence of the Newton-Raphson iterations occurs within a specified tolerance, if the solution is not accepted when maximum number of iterations is reached, and this error accumulates along time steps, leading eventually in large errors in the final response. Exception is the elastoplastic oscillator, in which energy is supposed to be dissipated due to the fact that its force – displacement diagram forms a closed loop as will be shown in later sections. The energy dissipation in each cycle is equal to the area of this loop.

In Figure 2, phase portraits of the response of the SDOF oscillator with hardening spring are shown with the 13 different integration methods mentioned above. It is noted that in the cases of the U0-V1-Opt, the U0-V1-CA and the U0-V1-DA algorithms, their integration parameters have been selected so that they actually correspond to the generalized a-method (denoted by CH-a in the appropriate subplot title and referred in [15]), the HHT-a method (denoted by HHT-a in the appropriate subplot title and referred in [16]) and the Wood-Bossak-Zienkiewicz method (denoted by WBZ-a in the appropriate subplot title and referred in [17]) respectively. It is clearly seen that the optimal numerical dissipation and dispersion methods (…-Opt) preserve the total energy much better that their corresponding continuous acceleration (…-CA) and discontinuous acceleration (…-DA) methods.

In Figure 3, phase portraits of the response of the SDOF oscillator with softening spring are shown with the 13 different integration methods. Observations analogous to those of Figure 2 can be made, since the optimal numerical dissipation and dispersion methods (…-Opt) lead to a much lower total energy decrease compared to the other methods. The decrease of total energy is more pronounced for all the time integrators except for the optimal ones, in the case of the softening spring, for which it has to be mentioned that the time step is 10 times higher than the one used for the hardening spring.

Finally, in Figure 4 the corresponding phase portraits are shown for the elastoplastic SDOF system. It is observed that except for the optimal numerical dissipation and dispersion algorithm with zero displacement and velocity overshooting (U0-V0-Opt), the phase portrait of which can be easily observable, all the other algorithms exhibit numerical instabilities. This is clear in the cases of U0-V1-Opt (CH-a) and U1-V0-Opt algorithms. Furthermore, apart from the three aforementioned algorithms, in all the other cases the solution is observed to be in error from the start of the dynamic analysis; as the displacement increases towards the positive direction, the velocity becomes negative and decreases. Taking into account the initial conditions of the elastoplastic oscillator, where the initial displacement and the initial velocity are both zero, this is not possible. Therefore, it is actually shown that the U0-V0-Opt algorithm performs best, compared to the others.

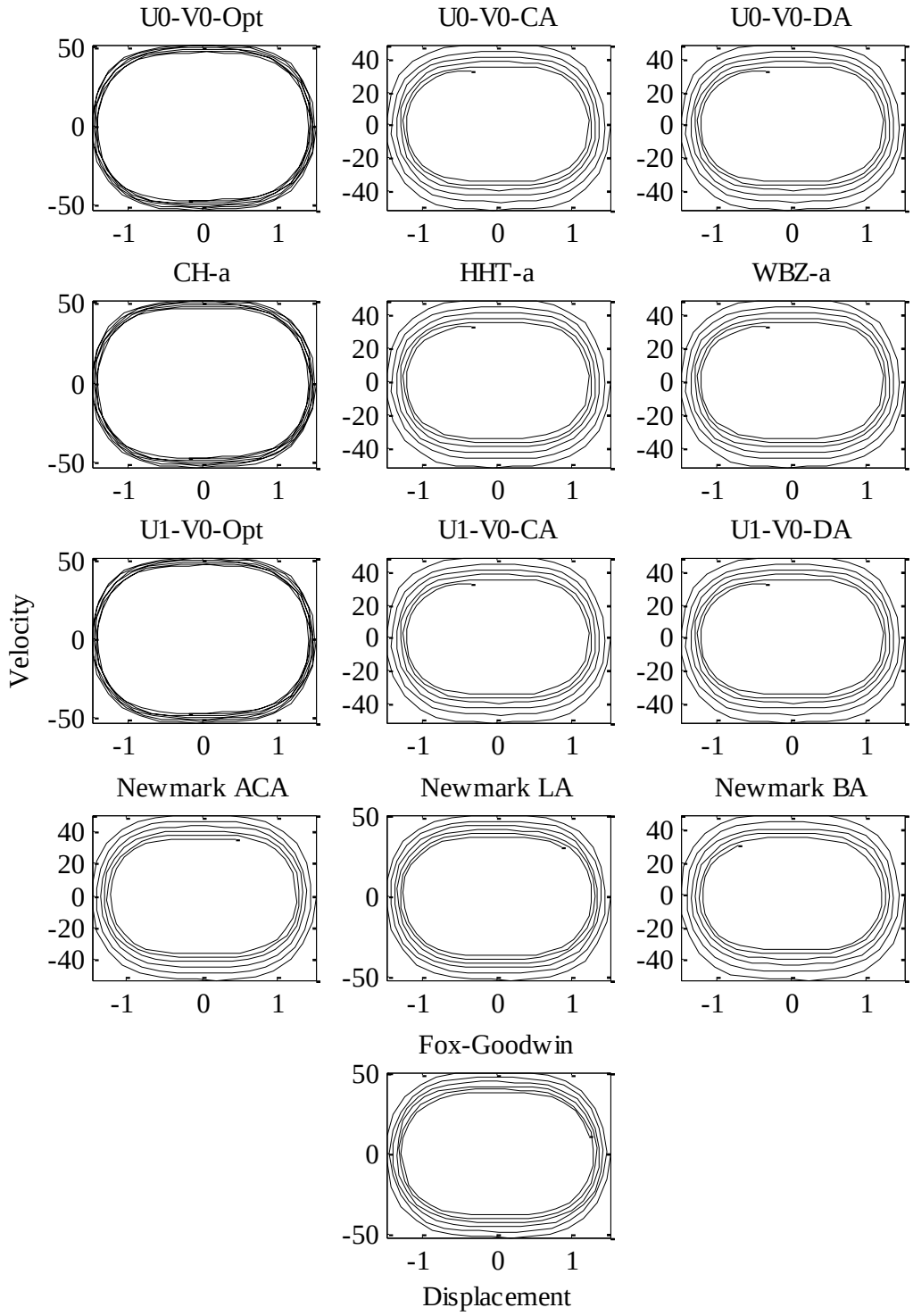

Figure 2: Phase portraits of the hardening spring $d^2u/dt^2+100u(1+10u^2)=0$, $(u)_0=1.5$, $(du/dt)_0=0$, $\Delta t=0.005$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

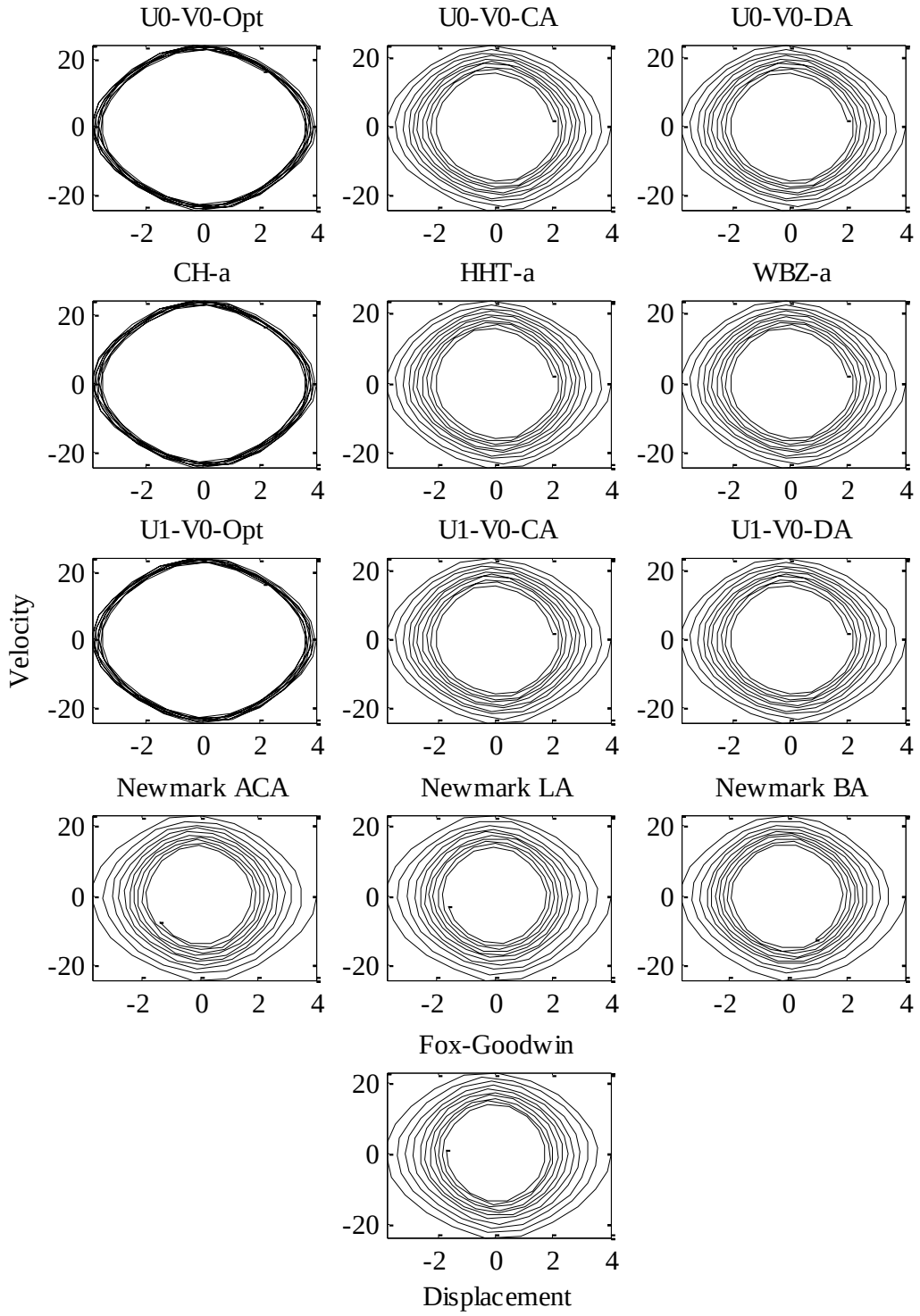

Figure 3: Phase portraits of the softening spring $d^2u/dt^2+100\tanh(u)=0$, $(u)_0=4$, $(du/dt)_0=0$, $\Delta t=0.05$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

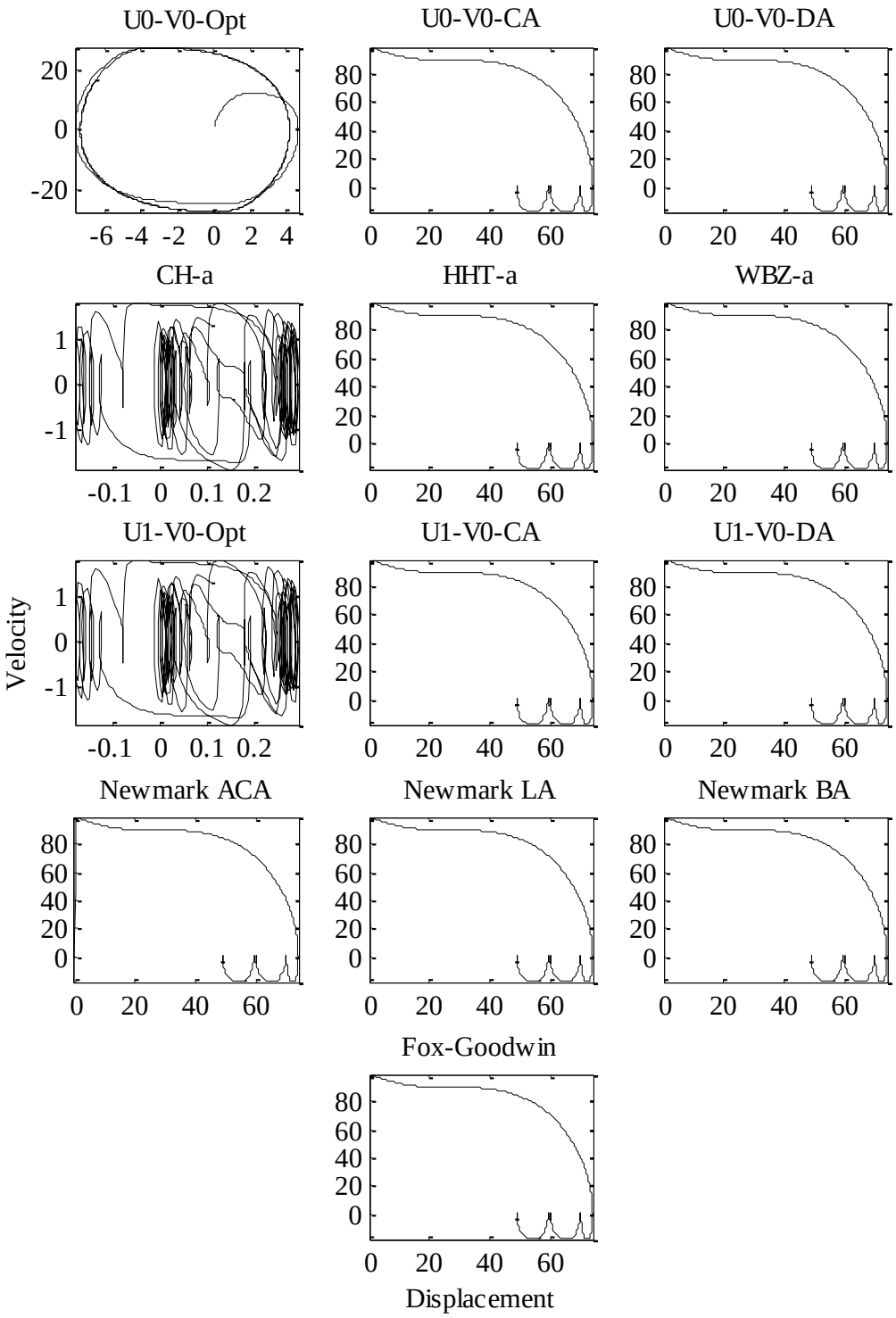

Figure 4: Phase portraits of the harmonically excited elastoplastic SDOF system, $(u)_0=0$, $(du/dt)_0=0$, $\Delta t= 0.005$, $t= 800\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

5.2 Energy variation

The energy variation for the various numerical integrators is shown in Figure 5 for the oscillator with the hardening spring, in Figure 6 for the oscillator with the softening spring and in Figure 7 for the oscillator with the elastoplastic spring. In each of these figures, the numerical integrators used are grouped into three categories, which are the optimal numerical dissipation and dispersion methods (represented with the continuous bold line), the continuous acceleration and discontinuous acceleration methods (represented with the dashed bold line) and finally the family of Newmark methods and the Fox-Goodwin method (represented with the continuous thin lines). The integration constants and other parameters are the same with those used for the results shown in Figures 1 to 3 respectively. The ideal time integration scheme would conserve energy in the first two oscillators, namely the energy deviation shown in Figures 4 and 5 would be zero. Among the algorithms studied here, energy reduction is present to a higher or lower extent. The lowest energy deviation is observed for the optimal numerical dissipation and dispersion algorithms. For the last, the energy reduction is about one third of that observed for continuous/discontinuous acceleration algorithms and the other common integration methods, in the case of the SDOF system with hardening spring (Figure 5). The reduced energy deviation for optimal numerical dissipation and dispersion algorithms is more pronounced in the case of the SDOF system with softening spring, where it is approximately one fourth of that observed for the remaining integration algorithms, as seen in Figure 6. In both Figures, the rate of decrease of total energy becomes maximum at the first few cycles of dynamic response for all the time integrators used.

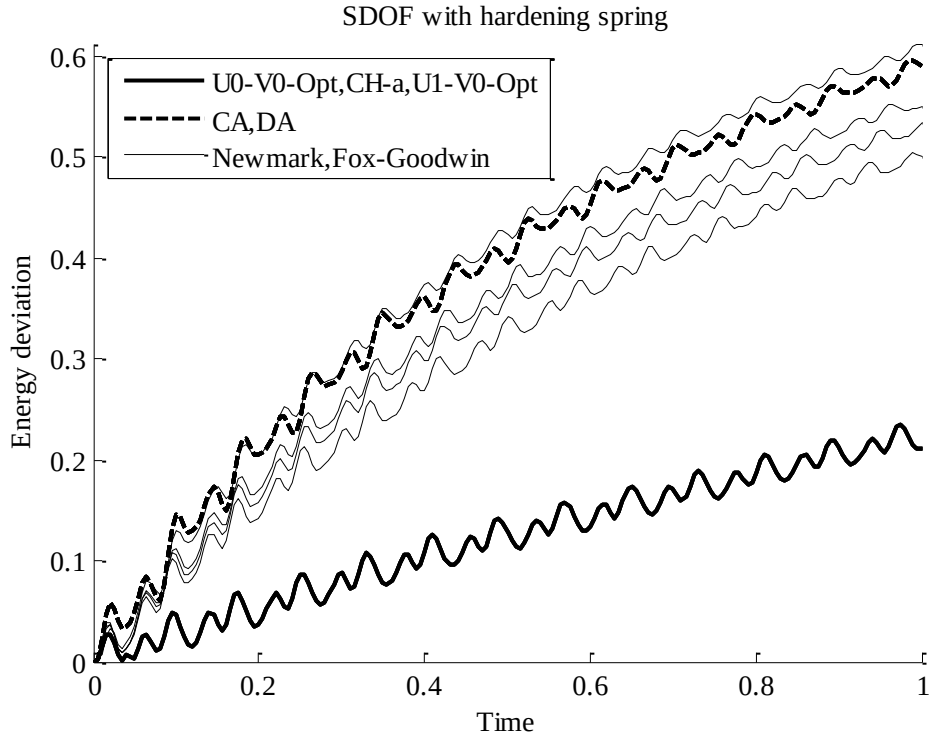

Figure 5: Total energy relative error of the SDOF oscillator with hardening spring $d^2u/dt^2+100u(1+10u^2)=0$, $(u)_0=1.5$, $(du/dt)_0=0$, $\Delta t=0.005$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

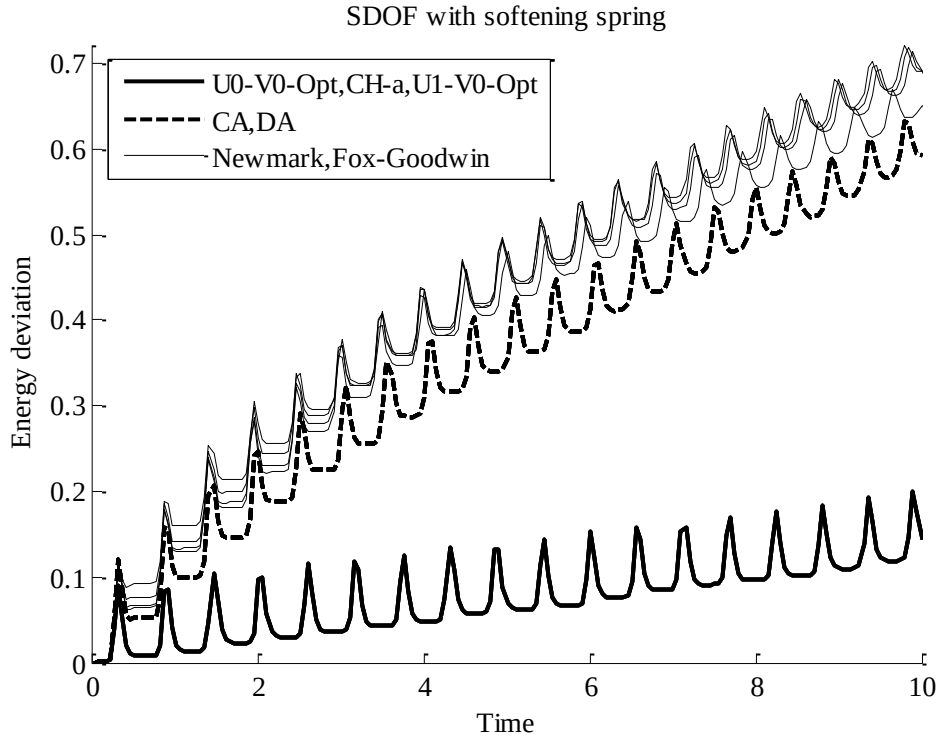

Figure 6: Total energy relative error of the SDOF oscillator with softening spring $d^2u/dt^2+100\tanh(u)=0$, $(u)_0=4$, $(du/dt)_0=0$, $\Delta t=0.05$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

In the case of the elastoplastic SDOF system, instead of the relative energy error, the value of the energy decay is plotted in Figure 7, since this system is not supposed to be conservative and its total energy variation cannot be analytically calculated. It is seen that the energy decay for the U0-V0-Opt algorithm is initially zero and gradually increases, with periodic undulations but with constant average slope. For the other two optimal numerical dissipation and dispersion methods (CH-a and U1-V0-Opt), there is a steep decrease of the energy decay from zero towards a large negative value, which means that at the initial time steps the spring does not dissipate energy as it should, but it increases the internal energy of the system, a fact that is clearly impossible. The same holds for the other algorithms used; this is a clear indication that these algorithms fail to simulate the dynamic response of the system, at least with the selected time step, while the U0-V0-Opt algorithm can efficiently integrate the differential equation of motion with increased accuracy and stability.

5.3 Number of iterations needed

The robustness of the U0-V0-Opt algorithm can be also deduced by comparison of the iterations needed by each time integration scheme to achieve convergence for the three types of oscillators used in this study. In Figure 8 the number of iterations for each of the 13 different time integration algorithms used in this study versus time is presented, for the SDOF system with the hardening spring. The corresponding results for the SDOF system with the softening spring and the SDOF system with the elastoplastic spring are presented in Figure 9 and Figure 10 respectively. The maximum number of iterations is equal to $k_{\max} = 200$. The integration constants and other parameters are identical to those used for the results presented in Figures 1-6 respectively for hardening, softening and elastoplastic SDOF systems. It is apparent that the optimal numerical dissipation and dispersion algorithms need an average of 10 iterations and a maximum of roughly 15 iterations to converge to the nonlinear solution at each time

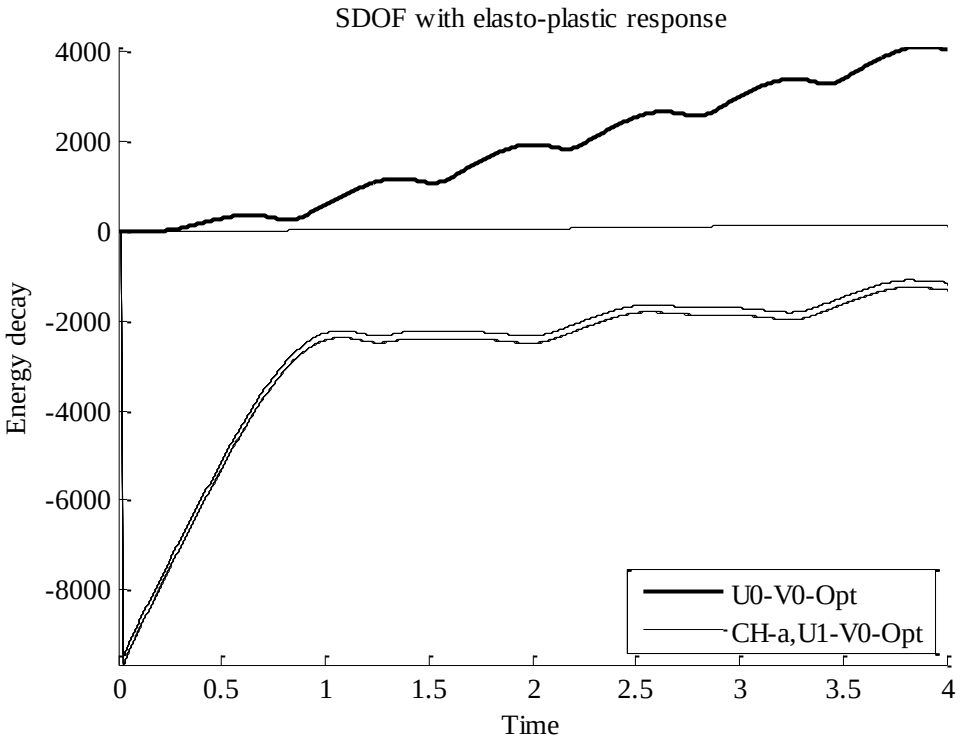

Figure 7: Total energy error of the harmonically excited elastoplastic SDOF system, $(u)_0=0$, $(du/dt)_0=0$, $\Delta t=0.005$, $t=800\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

step for the two first types of oscillators considered, whereas for the third type of oscillator only the U0-V0-Opt algorithm achieves convergence with a maximum of 20 iterations throughout the duration of the analysis. In contrast, the remaining algorithms are shown to reach the maximum iteration limit repetitively, at which time steps the solution is accepted without further check for equilibrium in terms of the dynamic equation of motion. This implies that for those algorithms much more error is accumulated to the calculated dynamic response of the oscillator, and consequently the numerical total energy decrease is higher. This error can be of course reduced by decreasing the time step; from another point of view this means that given an acceptable level of accuracy and stability, the U0-V0-Opt algorithm can be used with a larger time step than that used by the other integration algorithms, and therefore it is more numerically efficient, since it requires less computational effort for a given time span.

5.4 Internal force vs displacement diagrams

The computational energy loss during the dynamic analysis can be explained if the internal force – displacement diagram of the two types of oscillators studied here is plotted for the various time integration algorithms used. Such plots are shown in Figure 11 for the SDOF system with hardening spring, in Figure 12 for the SDOF system with softening spring and in Figure 13 for the SDOF system with the elastoplastic spring. It is obvious that the energy loss occurs due to the fact that in the internal force – displacement graphs a loop is formed between loading and unloading branches. The area inside the loop is equal to the amount of energy lost during an oscillation cycle. For the first and second oscillator types considered here, the loop is formed probably because of the numerical singularities existing at the maximum and minimum displacements from the equilibrium position. At these points the equilibrium path is totally reversed. On the other hand, the convergence tolerance and/or the maximum unsuccess

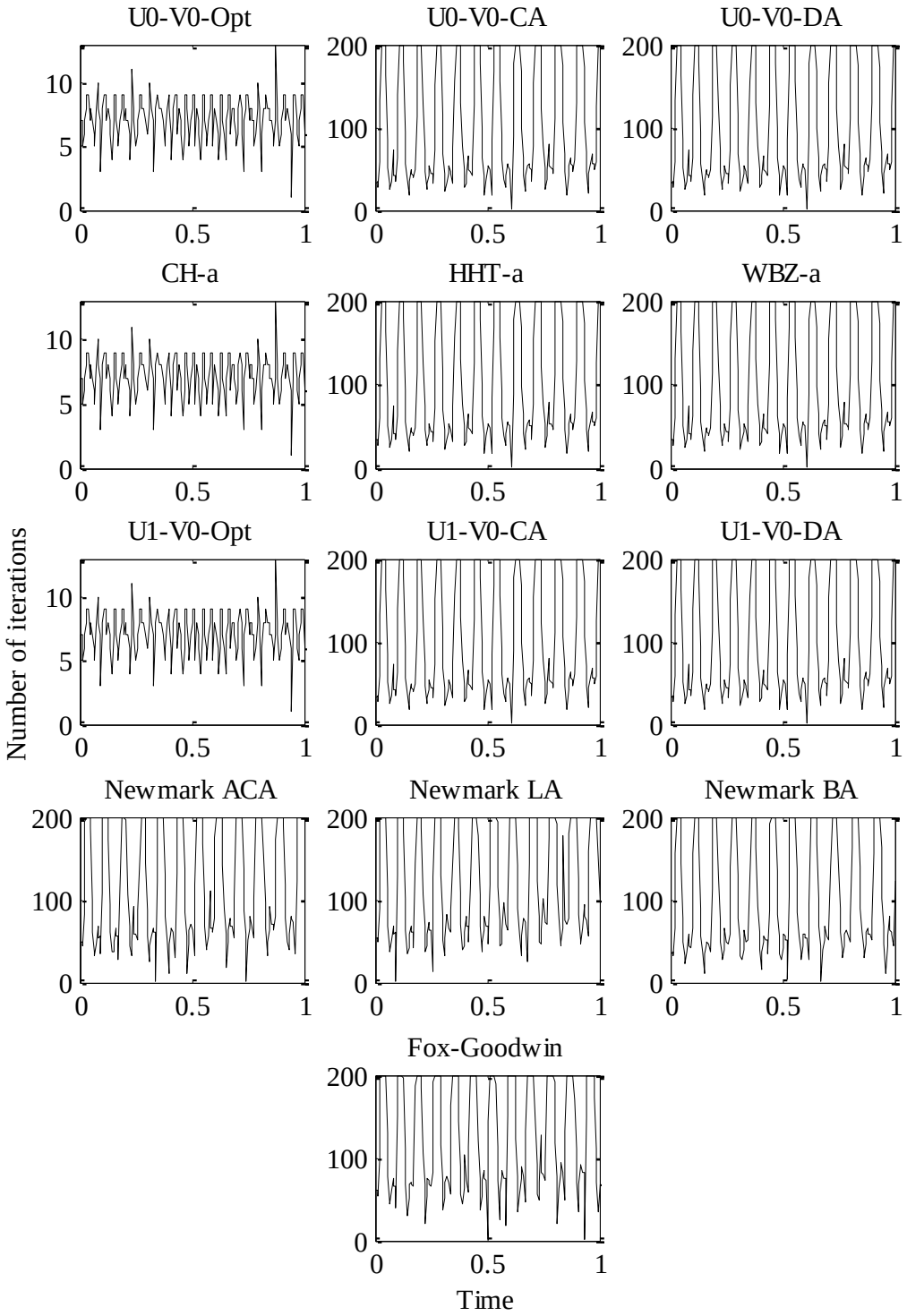

Figure 8: Number of iterations needed for the SDOF oscillator with hardening spring $d^2u/dt^2+100u(1+10u^2)=0$, $(u)_0=1.5$, $(du/dt)_0=0$, $\Delta t=0.005$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

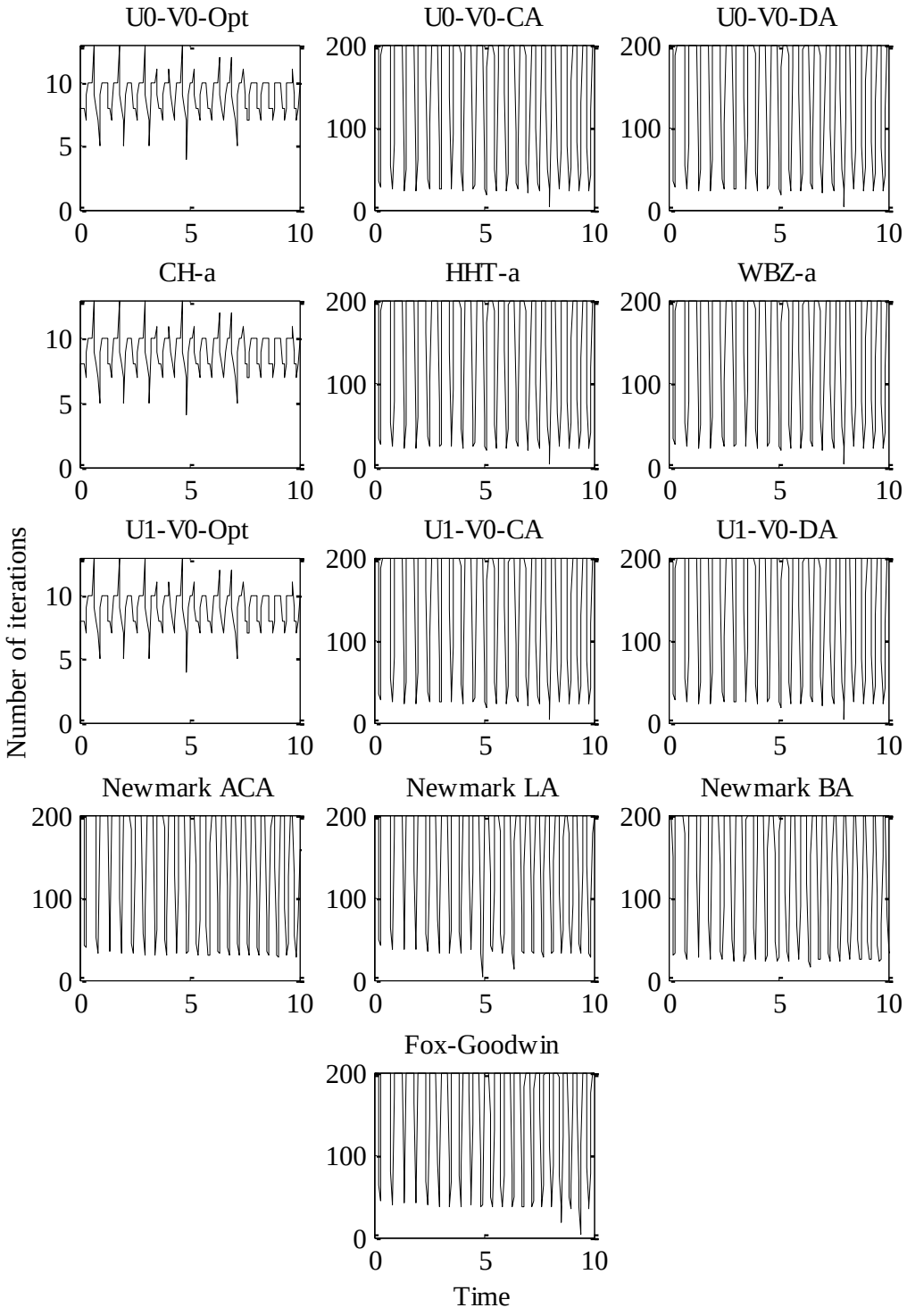

Figure 9: Number of iterations needed for the SDOF oscillator with softening spring $d^2u/dt^2+100\tanh(u)=0$, $(u)_0=4$, $(du/dt)_0=0$, $\Delta t=0.05$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

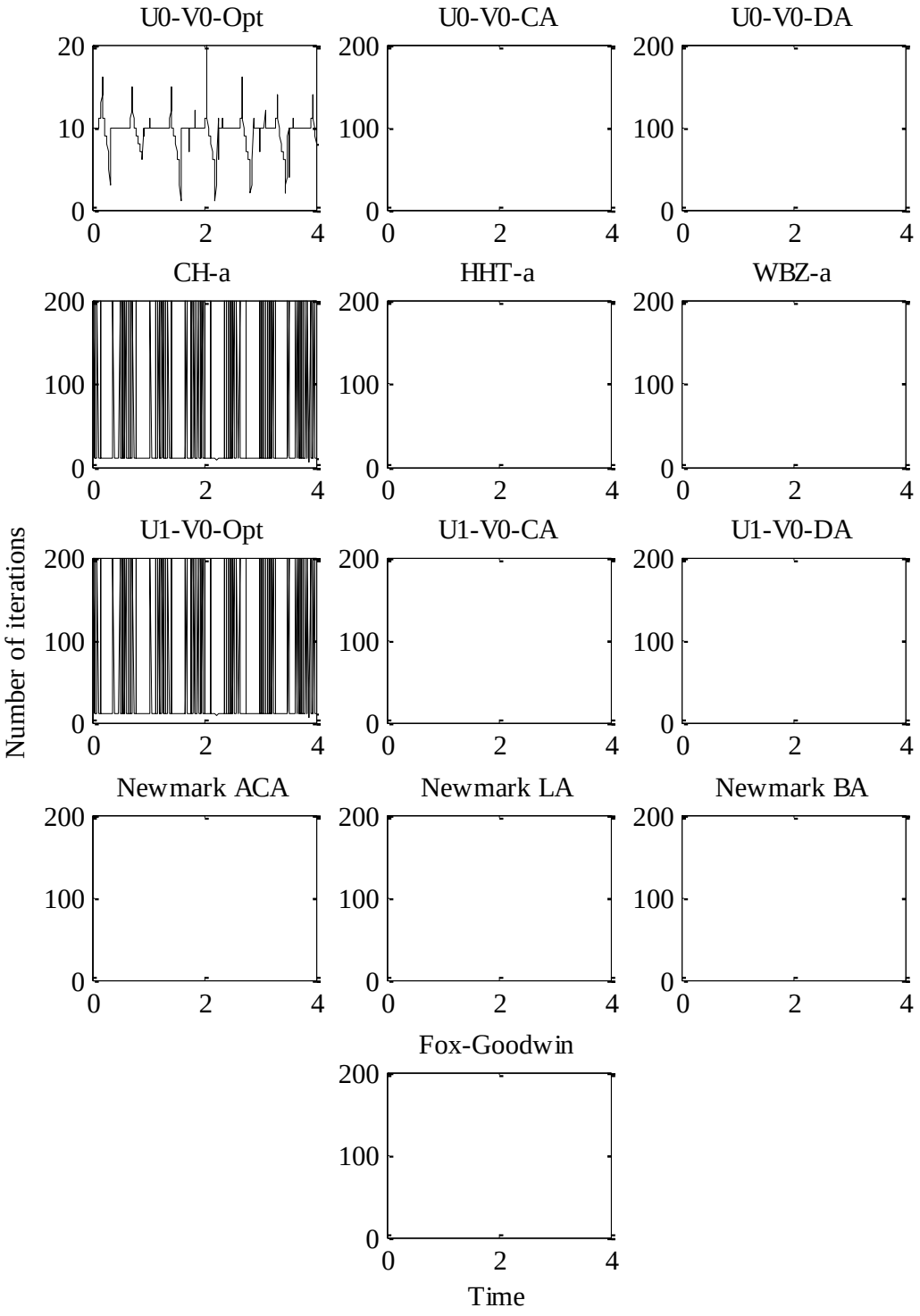

Figure 10: Number of iterations needed for the harmonically excited elastoplastic SDOF system, $(u)_0=0$, $(du/dt)_0=0$, $\Delta t= 0.005$, $t= 800\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

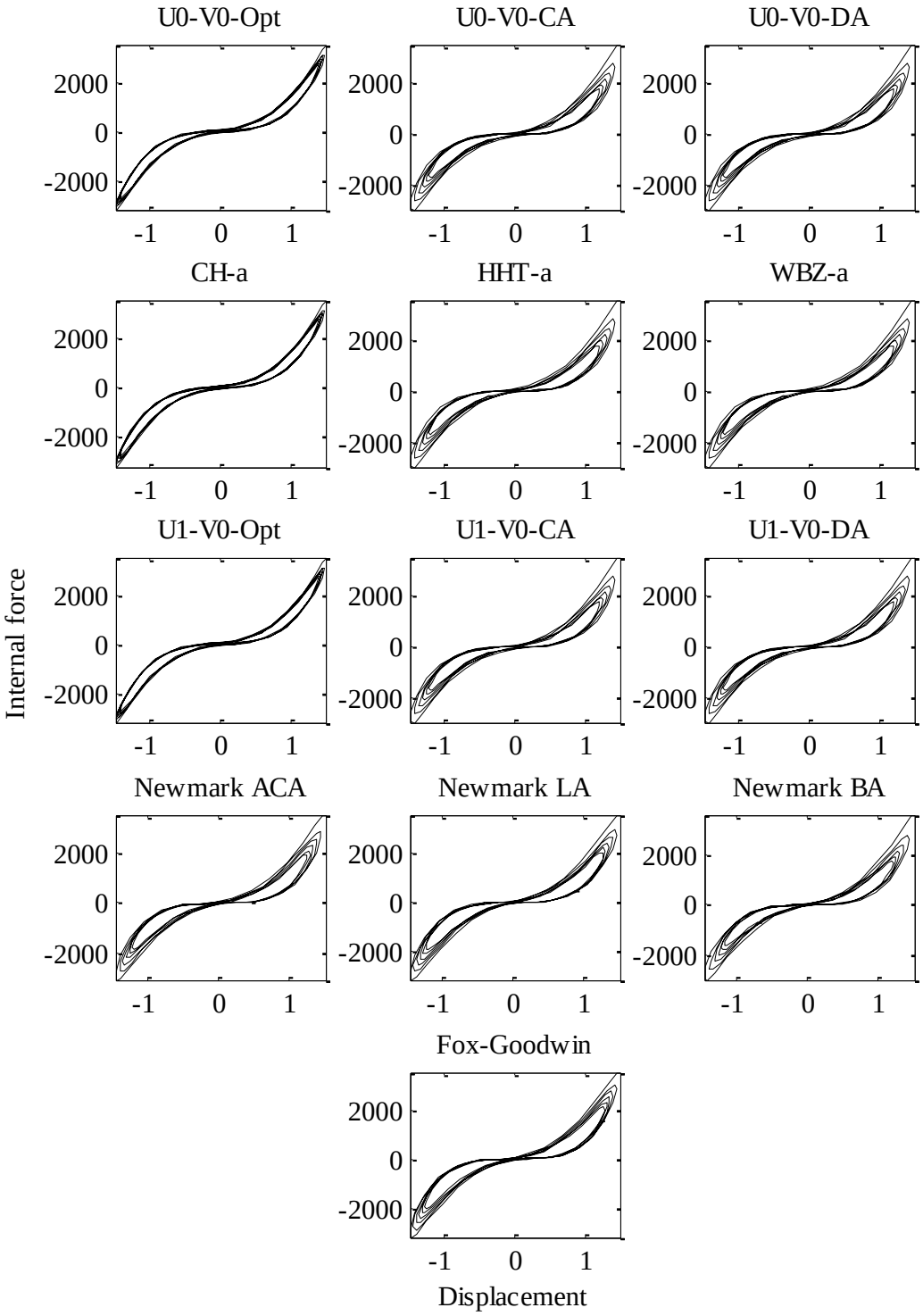

Figure 11: Internal force vs displacement curves for the hardening spring $d^2u/dt^2+100u(1+10u^2)=0$, $(u)_0=1.5$, $(du/dt)_0=0$, $\Delta t=0.005$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

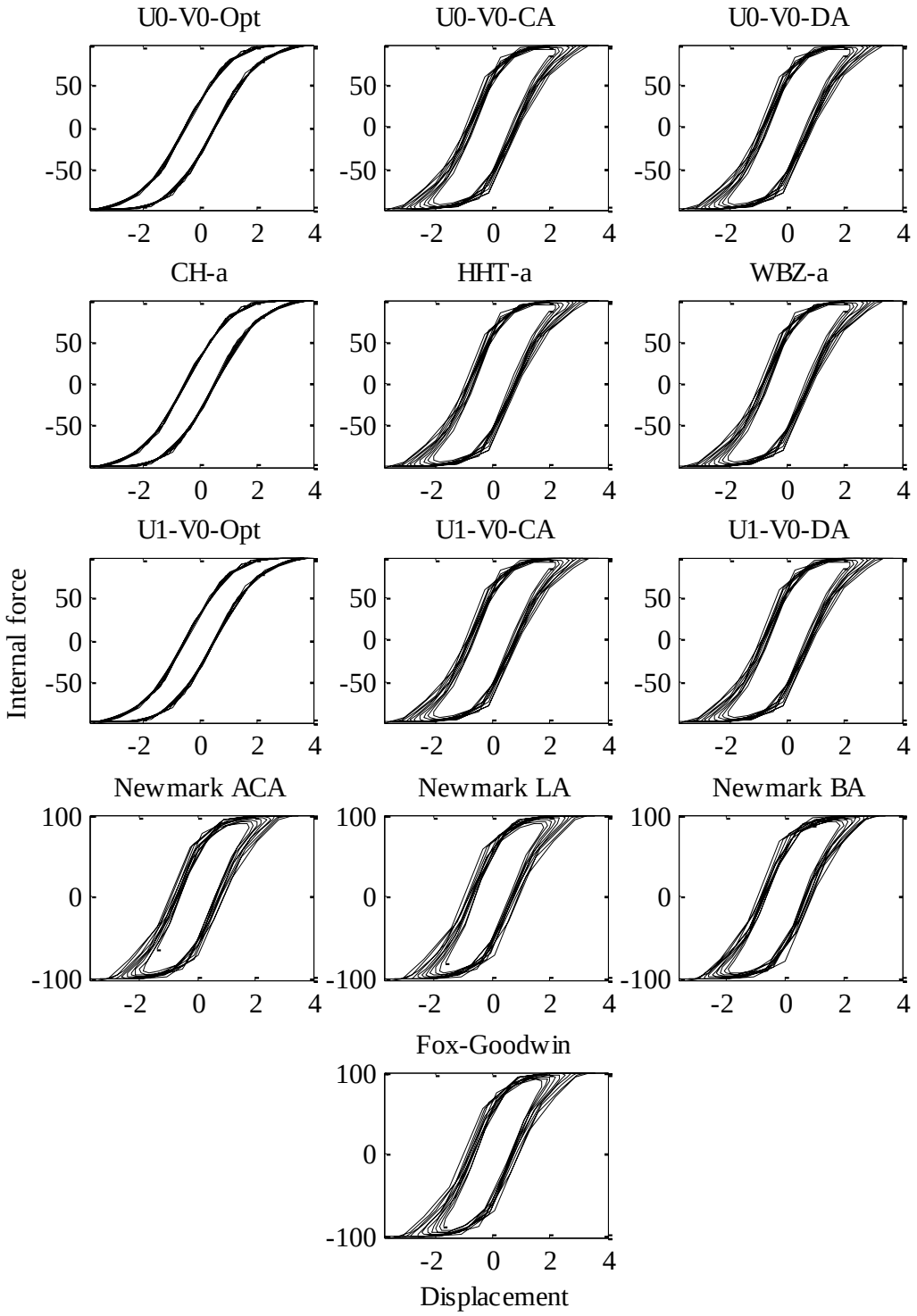

Figure 12: Internal force vs displacement curves for the softening spring $d^2u/dt^2+100\tanh(u)=0$, $(u)_0=4$, $(du/dt)_0=0$, $\Delta t=0.05$, $t=200\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

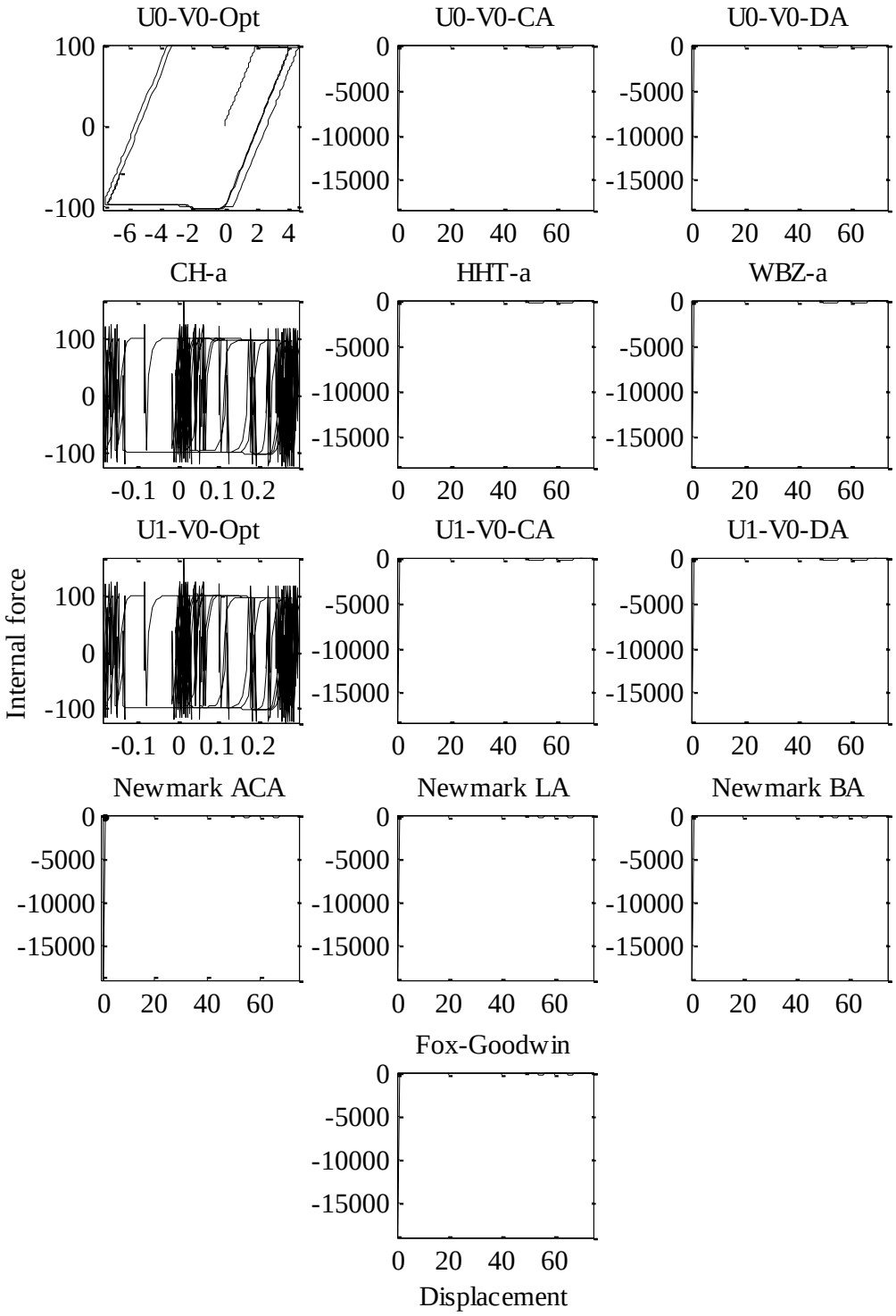

Figure 13: Internal force vs displacement curves for the harmonically excited elastoplastic SDOF system, $(u)_0=0$, $(du/dt)_0=0$, $\Delta t= 0.005$, $t= 800\Delta t$, $\rho_{inf}=1$, $tol_{max}=0.01$, $k_{max}=200$.

ful iterations reached, lead to errors in the calculation of the tangent stiffness, which are responsible for the different slopes between loading and unloading branches. For the third oscillator type the existence of the loop results from the elastoplastic behavior of the spring and is physically correct.

In addition, the area inside the loops is minimum for the optimal numerical dissipation and dispersion algorithms, compared to the remaining ones for the two first types of oscillators. Apart from this, the loop areas are generally larger for the SDOF system with softening spring, which means that the algorithmic energy dissipation will be higher for this type of oscillator. This is mainly due to the higher initial total energy in the SDOF system with softening spring compared to that of the SDOF system with hardening spring, since the relative energy loss observed in Figures 4 and 5 is roughly the same for the two oscillators.

The closed force – displacement loop for the elastoplastic behavior can be clearly seen in Figure 13 in the case of U0-V0-Opt algorithm. It is verified that the yield limit is equal to its assumed value for both tension and compression, and that the modulus of elasticity in the linear elastic range is equal to 50. Apparent instabilities exist for the remaining algorithms, which corroborate the superiority of U0-V0-Opt algorithm.

6 CONCLUSIONS

- The family of linear generalized single step single solve algorithms, of which most common time integration algorithms are special cases, can be extended to solve materially nonlinear dynamic response via a Newton-Raphson iterative procedure.
- In the nonlinear regime, the extended generalized algorithms perform satisfactorily, with acceptable accuracy and stability, in cases where the remaining common algorithms fail to trace the dynamic response.
- The algorithm which has zero-order displacement overshooting behavior, zero-order velocity overshooting behavior and optimal numerical dissipation and dispersion (U0-V0-Opt) can deal with nonlinear elastic as well as nonlinear elastic-plastic response with a relatively large time step.
- Time integration algorithms for which convergence or equilibrium is not ensured at time steps can lead to large errors in the calculated response. The relative energy decrease for conservative systems can be as much as 60% of its initial value.
- Algorithms that are unconditionally stable for linear problems, may lose their stability in nonlinear dynamic simulations.
- Further research has to be made to investigate the relation between the stable time increment of generalized single step single solve algorithms applied in nonlinear problems and various other input data.

REFERENCES

[1] X. Zhou, K.K. Tamma, Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. *Int. J. Numer. Meth. Engng*, **59**, 597–668, 2004.

[2] R.W. Clough, J. Penzien, *Dynamics of structures, 3th Edition*. Computers & Structures, Inc, 2003.

[3] J.H. Argyris, P.C. Dunne, T. Angelopoulos, Dynamic response by large step integration. *Earthquake Engineering & Structural Dynamics*, **2**, 185-203, 1973.

[4] D. Kuhl, M.A. Crisfield, Energy-conserving and decaying algorithms in non-linear structural dynamics. *Int. J. Numer. Meth. Engng*, **45**, 569-599, 1999.

[5] T.J.R. Hughes, T.K. Caughey, W.K. Liu, Finite-element methods for nonlinear elastodynamics which conserve energy. *Journal of Applied Mechanics, Transactions of the ASME*, **45**, 366-370, 1978.

[6] D. Kuhl, E. Ramm, Constraint energy momentum algorithm and its application to nonlinear dynamics of shells. *Computer Methods in Applied Mechanics and Engineering*, **136**, 293–315, 1996.

[7] J.C. Simo, N. Tarnow, The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *Journal of Applied Mathematics and Physics*, **43**(5), 757–792, 1992.

[8] K.J. Bathe, Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Computers and Structures*, **85**(7-8), 437–445, 2007.

[9] L. Zhang, T. Liu, Q. Li, A robust and efficient composite time integration algorithm for nonlinear structural dynamic analysis. *Mathematical Problems in Enginnering*, *Hindawi Publishing Corporation*, 2014.

[10] M.B. Rosales, C.P. Filipich, Time integration of non-linear dynamic equations by means of a direct variational method. *Journal of Sound and Vibration*, **254**(4), 763–775, 2002.

[11] F. Armero, E. Petöcz, A new class of conserving algorithms for dynamic contact problems. J.A. Désidéri, P. LeTallec, E. Oñate, J. Périaux, E. Stein eds. *2nd ECCOMAS Conf. on Numerical Methods in Engineering*, Paris, France, September 9-13, 1996.

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

[13] U.M. Ascher, L.R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*, SIAM, 1998.

[14] L. Fox, E.T. Goodwin, Some new methods for the numerical integration of ordinary differential equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, **45**, 373–388, 1949.

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

[16] H.M. Hilber, T.J.R. Hughes, R.L. Taylor, Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, **5**, 283-292, 1977.

[17] W.L. Wood, M. Bossak, O.C. Zienkiewicz, An alpha modification of Newmark's method. *International Journal for Numerical Methods in Engineering*, **15**(10), 1562-1566, 1980.

[18] Y.M. Xie, An assessment of time integration schemes for non-linear dynamic equations. *Journal of Sound and Vibration*, **192**(1), 321-331, 1996.

[19] J. Liu, X. Wang, An assessment of the differential quadrature time integration scheme for nonlinear dynamic equations. *Journal of Sound and Vibration*, **314**, 246-253, 2008.