REPORT NO.
UCB/SEMM-88/07

STRUCTURAL ENGINEERING,
MECHANICS AND MATERIALS

HIGHER DERIVATIVE EXPLICIT ONE STEP METHODS FOR NONLINEAR DYNAMIC PROBLEMS

BY

CLAUS HOFF and ROBERT L.TAYLOR

SEPTEMBER 1988

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

HIGHER DERIVATIVE EXPLICIT ONE STEP METHODS FOR NONLINEAR DYNAMIC PROBLEMS

C. Hoff[#] *and R. L. Taylor*[*]

ABSTRACT

For the solution of a certain class of nonlinear problems in structural dynamics a new higher order explicit one step scheme is presented. The generalized scheme covers some recently developed multistep algorithms as well as some new methods with improved numerical properties. A weighted residual approach is used which contains new aspects in the design of single step algorithms. The accuracy and stability are investigated theoretically for linear and nonlinear systems. Algorithms with higher accuracy than the central difference method are developed with the goal to pay as less as possible in the stability limit. The convergence is checked through practical calculations on linear and nonlinear systems. The increase of accuracy costs only a slight amount of computational effort. It is worthwhile to use these algorithms especially in weak and smooth nonlinearities.

Table of Contents

Presently visiting Structural Engineering, Mechanics and Materials, Department of Civil Engineering, University of California, Berkeley, CA 94720

* Professor, Structural Engineering, Mechanics and Materials, Department of Civil Engineering, University of California, Berkeley CA 94720

REPORT NO.
UCB/SEMM-88/07

STRUCTURAL ENGINEERING,
MECHANICS AND MATERIALS

HIGHER DERIVATIVE EXPLICIT ONE STEP METHODS FOR NONLINEAR DYNAMIC PROBLEMS

BY

CLAUS HOFF and ROBERT L.TAYLOR

SEPTEMBER 1988

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

HIGHER DERIVATIVE EXPLICIT ONE STEP METHODS FOR NONLINEAR DYNAMIC PROBLEMS

C. Hoff and R. L. Taylor

Department of Civil Engineering,
University of California, Berkeley

1. INTRODUCTION

To solve a nonlinear multidegree of freedom system in structural dynamics a step- by- step integration algorithm must be used. Practical applications and theoretical background of these algorithms can be found in the book of Belytschko and Hughes [1]. A state- of- the-art in step- by- step integration algorithms is recently summarized by Wood [2]. A lot of useful theoretical background exists in the mathematical literature, see e.g. [3]- [5]. Selecting an appropriate method for the nonlinear structural dynamic problem which is discretized in space, the following aspects must be considered :

- The totally discretized partial differential equation possesses always a truncation error composed of time and spatial approximation which can be roughly described by $r = O(\Delta t^k) + O(\Delta x^m)$ where Δt and Δx denote the time and spatial increment, respectively. Most of the spatial discretizations like finite differences or finite elements are not more than second order accurate in space ($m = 2$). If the time discretization leads to an accuracy $k > 2$, the spatial truncation error becomes dominant. Hence, the use of higher order in time pays off only if the cost is not significantly higher than for lower accurate time step schemes like central differences or Newmark methods. The main objective in developing higher order time step schemes for PDE's lies in the minimization of the additional effort compared to lower order algorithms.
- Higher order multistep methods are not always the best choice to increase the accuracy. They need special starting procedures, for example with Runge- Kutta methods. Some powerful multistep methods are inconvenient for special structural dynamic problems, for example the $A(\alpha)$- stable implicit methods restrict the time step in undamped systems.

- Higher order multistep methods maintain their accuracy only if the higher derivatives of the right hand side function fulfills some continuity requirements. When discontinuities occur the accuracy decreases strongly, see e.g. [5]. Plasticity, fracture mechanics and uplift problems are examples with discontinuities. It may be not worthwhile to use multistep methods with higher order in these problems. One step methods appear to overcome the obstacle of highly differentiable functions. But it will be shown in this paper that higher order one step methods possess of course the same drawback as multistep methods because they have to use higher derivatives to increase the accuracy.
- The transient problem in structural dynamics is almost classified as stiff. In [6] a stiff problem is described precisely. Following this definition the Lipschitz constant of the Jacobian must always be scaled with respect to the necessary time step. Impact and wave propagation problems require a small time step to get sufficient accuracy. Explicit conditionally stable methods suit in these less stiff cases. Inertia problems, for example in earthquake engineering, provide smooth solutions. Large time steps can be used and the problem must be classified as stiff. Implicit unconditionally stable methods with arbitrary large step size are the convenient choice in these cases.
- In nonlinear problems which are discretized by finite elements, finite differences or boundary elements the function evaluations (calculation of the internal forces) are the most expensive part. Thus, the usefulness of an algorithm is always measured with the required number of function evaluations. Higher order extrapolation methods for example suffer from an amount of function evaluations especially when discontinuities occur, see [5].

These aspects may be some of the reasons why most of the computer programs in structural dynamics and most of the research groups in this field are working with one step methods of relatively low order.

Implicit integration methods must solve a linear matrix equation in each time step. Explicit methods can avoid this effort if a lumped mass and damping matrix is used. The explicit scheme consists of pure vector operations which is very convenient for vector processors. Implicit algorithms are able to reach unconditional stability with arbitrary large time step sizes. Explicit methods are always conditionally stable and the time step size is restricted by the highest eigenfrequency of the system. This disadvantage can be partially

reduced through a consequent vectorized implementation. Therefore explicit algorithms become more and more competitive to implicit algorithms.

In the last years several developments were made to improve explicit methods. The second order accurate central difference method is the most popular explicit algorithm, applications can be found in [1]. In 1970 Fu [7] presented an explicit two step method which is fourth order accurate and has a lower stability limit than the central difference method. In 1984 Kujawski and Gallagher [8] developed a new family of two step Newmark algorithms with more than one function evaluation. They found a fourth order algorithm with a stability limit which is higher than that of Fu's method [7] and lower than that of the central difference method. In addition Kujawski and coworkers discovered some second and fourth order versions with more function evaluations, see [9] and further references. The performance of the new algorithms was shown for mildly nonlinear systems. In 1985 Katona and Zienkiewicz [10] developed some higher order explicit one step methods. They are easy to implement in the nonlinear case and the higher order is reached to the cost of only a few more vector operations which is nothing compared to function evaluations or matrix operations. The stability limit of these methods is slightly lower compared to the central difference method, see table IX in [10]. The application of these algorithms to nonlinear systems is not published as yet. These recent results represent the state of- the- art in higher order explicit one step schemes in structural dynamics.

After some remarks to the exact solution we start with a new concept in the design of step- by- step integration methods. From the resulting generalized single step algorithm several higher order explicit methods are derived with the objective to get the stability limit and the accuracy as high as possible. Accuracy and stability is theoretically proved for linear undamped systems and then extended to nonlinear systems. The performance of the algorithms is shown for linear and nonlinear systems.

2. GUIDELINES TO INCREASE THE ACCURACY OF STEP- BY- STEP INTEGRATION METHODS

2.1 Remarks to the Exact Solution

We want to solve the nonlinear equation of motion with initial conditions :

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

$$\mathbf{u}(0) = \mathbf{u}_0$$

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

with the mass matrix $\mathbf{M}$, the vector of the nonlinear internal forces $\mathbf{f}(\mathbf{u}(t),\dot{\mathbf{u}}(t))$, the vector of the external loads $\mathbf{p}(t)$ and the displacement vector $\mathbf{u}(t)$. The dot denotes differentiation with respect to time.

For the following development of a generalized step- by- step integration algorithm the second order differential equation (1) is transformed in two first order differential equations by introducing a new variable $\mathbf{w} = \dot{\mathbf{u}}$ for the velocities :

$$\dot{\mathbf{u}} = \mathbf{w} \tag{2}$$

$$\mathbf{M}\,\dot{\mathbf{w}} = \mathbf{p}(t) - \mathbf{f}(\mathbf{u}(t),\mathbf{w}(t))$$

The problem (2) can now be written as a first order differential equation:

$$\dot{\mathbf{y}}(t) = \mathbf{g}(\mathbf{y}(t)) + \mathbf{q}(t) = \mathbf{h}(\mathbf{y}(t),t) \tag{3}$$

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{w}(t) \end{bmatrix} \qquad \mathbf{g}(\mathbf{y}(t)) = \begin{bmatrix} \mathbf{w}(t) \\ -\mathbf{M}^{-1}\,\mathbf{f}(\mathbf{u},\mathbf{w}) \end{bmatrix} \qquad \mathbf{q} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\,\mathbf{p}(t) \end{bmatrix}$$

Some theoretical considerations are made on the linear case:

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{C}\,\dot{\mathbf{u}}(t) + \mathbf{K}\,\mathbf{u}(t) = \mathbf{p}(t) \tag{4}$$

with the corresponding first order version

$$\dot{\mathbf{y}}(t) = \mathbf{B}\,\mathbf{y}(t) + \mathbf{q}(t) \tag{5}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\,\mathbf{K} & -\mathbf{M}^{-1}\,\mathbf{C} \end{bmatrix}$$

The corresponding eigenvalue problem of the homogeneous part of (5)

$$\mathbf{B}\,\mathbf{z}_j = \phi_j\,\mathbf{z}_j \qquad j = 1, \cdots 2N \tag{6}$$

leads to 2N eigenvalues if we have N degrees of freedom in the original problem (4). Assuming modal damping in the undercritical range we get 2N complex conjugate eigenvalues. For the single degree of freedom system we obtain

$$\phi_{1,2} = -\xi\,\omega \pm i\omega_D \tag{7}$$

$$\omega_D = \omega\,(1-\xi^2)^{1/2}\ , \qquad \omega = \left(\frac{K}{M}\right)^{1/2}\ , \qquad \xi = \frac{C}{2\,(K\,M)^{1/2}}\ , \qquad i = (-1)^{1/2}$$

Starting with the known initial conditions (1b) the exact solution of (5) can theoretically be developed from the time step t_n to the next step t_{n+1} over the interval Δt :

$$\mathbf{y}_{n+1} = e^{\mathbf{B}\,\Delta t}\ \mathbf{y}_n + \mathbf{r}_{n+1} \tag{8}$$

$$\mathbf{r}_{n+1} = e^{\mathbf{B}\,t_{n+1}} \int_{t_n}^{t_{n+1}} e^{-\mathbf{B}\,t}\,\mathbf{q}(t)\,dt$$

The numerical method has to copy the exact solution (8) as close as possible. Only the displacements and the velocities are necessary to go from one time step to the other. With respect to the exact eigenvalues (7) the amplification matrix of the step- by- step integration method must consist of two complex conjugate roots. Several methods possess this highly desired property, for example the trapezoidal rule of Newmark and the central difference method, see [1], the higher order method of Gellert [11], some versions of the generalized algorithms of Zienkiewicz, Wood, Hine and Taylor [12] and of Katona [10] and the new higher order algorithms of Kujawski and Gallagher [8].

To filter numerical noise it may be advantageous to have a third root in the higher frequency range. In implicit unconditionally stable methods with large time steps the α-method of Hilber [13] and the Θ_1 - method [14], [15] use a third root to introduce numerical dissipation. In explicit methods numerical dissipation is rather necessary because the time step is restricted by the stability limit. To smooth the numerical solution it may be useful to introduce numerical dissipation even for explicit schemes. This will be figured out in the following chapters.

2.2 Possibilities to Increase the Accuracy

In general the following two possibilities exist to increase the accuracy over $O(\Delta t^2)$ for the nonlinear problem (1) or (3):

- Higher order linear multistep schemes with two or more steps or the equivalent one step schemes based on Taylor series in **y** are able to increase the order. The methods are called linear as long as the resulting difference equation is a linear combination of the form:

$$\sum_{i=0}^{k} \alpha_i \, \mathbf{y}_{n+i} \;=\; \Delta t \sum_{i=0}^{k} \beta_i \, \mathbf{h}_{n+i} \tag{9}$$

  Various methods exist for the first and second order differential equation (1) or (3). The linear multistep methods are in general based on Adam's type methods, see chapter III in [3]. With the use of Taylor series in the approximation functions of **y**(t) it is always possible to construct a one step method which is spectral identical to the corresponding multistep method. This idea of Taylor series expansions is due to Nordsieck, see chapter III in [3] and was recently improved by Corliss and Chang [16]. The developments in [14] and [15] and the β_m- family of Katona and Zienkiewicz [10] can be classified as such a linear one step method with Taylor series expansions in the approximation function for **y**(t).

- The second possibility to increase the order is provided by nonlinear schemes which need sometimes less steps than linear schemes. The Runge- Kutta, Nyström and extrapolation methods can be classified as nonlinear, see chapter II in [3]. Further alternatives are the higher derivative methods and the related Padé approximant methods. The fundamental work was made by Nørsett [17] and Enright [18], [19] and was developed further by Sacks- Davis and Shampine [20]. The family of second derivative methods are defined by the following formula:

$$\mathbf{y}_{n+1} \;=\; \mathbf{y}_n + \Delta t \sum_{i=0}^{k} b_i \, \dot{\mathbf{y}}_{n+1-i} \;+ \Delta t^2 \sum_{j=0}^{l} c_j \, \ddot{\mathbf{y}}_{n+1-j} \tag{10}$$

  If the function **h**(**y**(t),t) is linear in **y** that means the case (4), (5) the methods defined by (10) are related to the so called Padé approximant methods, see Addison and Gladwell [21] and Thomas and Gladwell [22]. They are based on the approximation of the matrix exponent in (8), see Lancaster [23] for the theoretical background.

$$e^{\mathbf{B}\,\Delta t} \;\approx\; \mathbf{I} + \Delta t\,\mathbf{B} + \frac{1}{2}\Delta t^2\,\mathbf{B}^2 + \frac{1}{6}\Delta t^3\,\mathbf{B}^3 + \ldots. \tag{11}$$

The resulting methods have the following scheme:

$$\mathbf{B}_0\, \mathbf{y}_{n+1} = \mathbf{B}_1\, \mathbf{y}_n + \mathbf{r}_n \tag{12}$$

In the method of Gellert [11] for example the matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ are constructed from the following formulas:

$$\mathbf{B}_0 = \mathbf{I} - \frac{\Delta t}{2}\mathbf{B} + \frac{\Delta t^2}{12}\mathbf{B}^2 \tag{13}$$

$$\mathbf{B}_1 = \mathbf{I} + \frac{\Delta t}{2}\mathbf{B} + \frac{\Delta t^2}{12}\mathbf{B}^2$$

Other methods in this class were developed by Argyris et. al. [24], Trujillo [25], Brusa and Nigro [26] and Serbin [27]. The following development can also be classified as a higher derivative method or in the linear case as a Padé approximant method. In contrast to the former investigations which were mainly focussed on implicit schemes the scope in the present work is concentrated on explicit schemes. The matrix $\mathbf{B}_0$ must become diagonal and matrix multiplications on the right hand side of (12) must be avoided in the implemented version.

3. DEVELOPMENT OF A NEW GENERALIZED ONE STEP ALGORITHM

Several useful approaches exist to design a step- by- step method from the principle of weighted residuals. Zienkiewicz, Wood and Taylor [28] and Wood [29] did a lot of research in this field. The recent attempt to unify most of the known algorithms can be found in the paper of Zienkiewicz, Wood, Hine and Taylor [12]. The generalized algorithm of Katona and Zienkiewicz [10] is a Taylor series approach which fits the algorithms in [12] in the sense of identical spectral properties. In [14] the first author made a different weighted residual approach with incompatible approximation functions. The following development starts with the first order equation (3). Penry and Wood proposed something similar in [30].

Starting from the first order form (5) we restrict the development to the linear case for simplicity and extend it to the nonlinear case later on. To take numerical advantage and to reflect the sense of the form (5) we introduce two independent sets of approximation functions for the displacements and the velocities. The functions are evaluated up to the

order three for the reason of simplicity. It is of course possible to extend the development to arbitrary high order. For the time incrementation the following notation is used:

$$\tau = t - t_n \ , \quad t_n \le t \le t_{n+1} \ , \quad 0 \le \tau \le \Delta t \ , \quad \Delta t = t_{n+1} - t_n \tag{14}$$

The displacements, velocities and its derivatives are developed in Taylor series :

$$\begin{aligned}
\mathbf{d}(\tau) &= \mathbf{d}_n + \dot{\mathbf{d}}_n \tau + \beta_2 \ddot{\mathbf{d}}_n \tau^2 + \beta_3 \frac{\Delta \ddot{\mathbf{d}}}{\Delta t} \tau^3 \\
\dot{\mathbf{d}}(\tau) &= \dot{\mathbf{d}}_n + \ddot{\mathbf{d}}_n \tau + \gamma_2 \frac{\Delta \ddot{\mathbf{d}}}{\Delta t} \tau^2 \\
\ddot{\mathbf{d}}(\tau) &= \ddot{\mathbf{d}}_n + \frac{\Delta \ddot{\mathbf{d}}}{\Delta t} \tau
\end{aligned} \tag{15}$$

$$\begin{aligned}
\mathbf{v}(\tau) &= \mathbf{v}_n + \dot{\mathbf{v}}_n \tau + \eta_2 \ddot{\mathbf{v}}_n \tau^2 + \eta_3 \frac{\Delta \ddot{\mathbf{v}}}{\Delta t} \tau^3 \\
\dot{\mathbf{v}}(\tau) &= \dot{\mathbf{v}}_n + \ddot{\mathbf{v}}_n \tau + \kappa_2 \frac{\Delta \ddot{\mathbf{v}}}{\Delta t} \tau^2 \\
\ddot{\mathbf{v}}(\tau) &= \ddot{\mathbf{v}}_n + \frac{\Delta \ddot{\mathbf{v}}}{\Delta t} \tau
\end{aligned} \tag{16}$$

The load function must also be approximated in a convenient way:

$$\mathbf{p}(\tau) = \mathbf{p}_n + \dot{\mathbf{p}}_n + \frac{1}{2} \ddot{\mathbf{p}}_n \tau^2 + \frac{1}{6} \frac{\Delta \ddot{\mathbf{p}}_n}{\Delta t} \tau^3 \tag{17}$$

The exact displacements $\mathbf{u}(\tau)$ and velocities $\mathbf{w}(\tau)$ in (5) are substituted by the approximations $\mathbf{d}(\tau)$ (15) and $\mathbf{v}(\tau)$ (16). The two first order differential equations are weighted by two independent weighting functions $w_1(\tau)$ and $w_2(\tau)$ over the time interval. The two differential equations have to be approximated in a weighted residual sense:

$$\frac{1}{\Delta t} \int_0^{\Delta t} [\, \mathbf{M} \dot{\mathbf{v}}(\tau) + \mathbf{C}\, \mathbf{v}(\tau) + \mathbf{K}\, \mathbf{d}(\tau) \,]\, w_1(\tau)\, d\tau \ = \ \frac{1}{\Delta t} \int_0^{\Delta t} \mathbf{p}(\tau)\, w_1(\tau)\, d\tau \tag{18}$$

$$\frac{1}{\Delta t} \int_0^{\Delta t} [\, \dot{\mathbf{d}}(\tau) - \mathbf{v}(\tau) \,]\, w_2(\tau)\, d\tau \ = \ 0 \tag{19}$$

To perform the integration the following integrals are defined:

$$\frac{1}{\Delta t}\int_0^{\Delta t} w_j(\tau)\, d\tau = 1 \qquad j = 1,2$$

$$\frac{1}{\Delta t}\int_0^{\Delta t} w_1(\tau)\, \tau^i\, d\tau = \Psi_i\, \Delta t^i \qquad i \geq 1 \tag{20}$$

$$\frac{1}{\Delta t}\int_0^{\Delta t} w_2(\tau)\, \tau^i\, d\tau = \Pi_i\, \Delta t^i \qquad i \geq 1$$

Inserting the approximation functions (15) and (16) into (18) and (19) and integrating with help of (20) we get two equations for the unknowns $\Delta\ddot{\mathbf{v}}$ and $\Delta\ddot{\mathbf{d}}$:

$$(\kappa_2\, \Psi_2\, \Delta t\, \mathbf{M} + \eta_3\, \Psi_3\, \Delta t^2\, \mathbf{C})\, \Delta\ddot{\mathbf{v}} = \tag{21}$$

$$\mathbf{p}_n + \Psi_1\, \dot{\mathbf{p}}_n\, \Delta t + \frac{1}{2}\Psi_2\, \ddot{\mathbf{p}}_n\, \Delta t^2 + \frac{1}{6}\Psi_3\, \Delta\ddot{\mathbf{p}}\, \Delta t^2$$

$$- \mathbf{M}\,(\dot{\mathbf{v}}_n + \Psi_1\, \ddot{\mathbf{v}}_n\, \Delta t)$$

$$- \mathbf{C}\,(\mathbf{v}_n + \Psi_1\, \dot{\mathbf{v}}_n\, \Delta t + \eta_2\, \Psi_2\, \ddot{\mathbf{v}}_n\, \Delta t^2)$$

$$- \mathbf{K}\,(\mathbf{d}_n + \Psi_1\, \dot{\mathbf{d}}_n\, \Delta t + \beta_2\, \Psi_2\, \ddot{\mathbf{d}}_n\, \Delta t^2 + \beta_3\, \Psi_3\, \Delta\ddot{\mathbf{d}}\, \Delta t^2)$$

$$\gamma_2\, \Pi_2\, \Delta\ddot{\mathbf{d}}\, \Delta t = \mathbf{v}_n - \dot{\mathbf{d}}_n + \Pi_1\, \dot{\mathbf{v}}_n\, \Delta t - \Pi_1\, \ddot{\mathbf{d}}_n\, \Delta t + \eta_2\, \Pi_2\, \ddot{\mathbf{v}}_n\, \Delta t^2 + \eta_3\, \Pi_3\, \Delta\ddot{\mathbf{v}}\, \Delta t^2 \tag{22}$$

The new state at the time t_{n+1} is calculated from (15) and (16) setting $\tau = \Delta t$ with the vectors $\Delta\ddot{\mathbf{v}}$ and $\Delta\ddot{\mathbf{d}}$ from (21) and (22):

$$\begin{aligned}
\mathbf{d}_{n+1} &= \mathbf{d}_n + \dot{\mathbf{d}}_n\, \Delta t + \beta_2\, \ddot{\mathbf{d}}_n\, \Delta t^2 + \beta_3\, \Delta\ddot{\mathbf{d}}\, \Delta t^2 \\
\dot{\mathbf{d}}_{n+1} &= \dot{\mathbf{d}}_n + \ddot{\mathbf{d}}_n\, \Delta t + \gamma_2\, \Delta\ddot{\mathbf{d}}\, \Delta t \\
\ddot{\mathbf{d}}_{n+1} &= \ddot{\mathbf{d}}_n + \Delta\ddot{\mathbf{d}}
\end{aligned} \tag{23}$$

$$\begin{aligned}
\mathbf{v}_{n+1} &= \mathbf{v}_n + \dot{\mathbf{v}}_n\, \Delta t + \eta_2\, \ddot{\mathbf{v}}_n\, \Delta t^2 + \eta_3\, \Delta\ddot{\mathbf{v}}\, \Delta t^2 \\
\dot{\mathbf{v}}_{n+1} &= \dot{\mathbf{v}}_n + \ddot{\mathbf{v}}_n\, \Delta t + \kappa_2\, \Delta\ddot{\mathbf{v}}\, \Delta t^2 \\
\ddot{\mathbf{v}}_{n+1} &= \ddot{\mathbf{v}}_n + \Delta\ddot{\mathbf{v}}
\end{aligned} \tag{24}$$

With (21) - (24) the generalized algorithm is completely described. It is of course a long

way to fit the various free time constant parameters to get a powerful method with all desired properties regarding accuracy, stability and performance in nonlinear problems.

Before we go into details it is worth to make some theoretical considerations and compare the generalized scheme with the requirements of the exact solution described in the foregoing chapter. If we eliminate the primary unknowns $\Delta\ddot{\mathbf{d}}$ and $\Delta\dot{\mathbf{v}}$ in (23) and (24) with the equations (21) and (22) we end up with a set of six state vectors compared to two necessary state vectors in the exact solution (8). In a single degree of freedom system the numerical scheme can be pressed into the following matrix vector notation :

$$\mathbf{x}_{n+1} = \mathbf{A}\,\mathbf{x}_n + \mathbf{q}_n \tag{25}$$

$$\mathbf{x}_n^T = [\, d_n,\ \dot{d}_n\,\Delta t,\ \ddot{d}_n\,\Delta t^2,\ v_n\,\Delta t,\ \dot{v}_n\,\Delta t^2,\ \ddot{v}_n\,\Delta t^3\,]$$

with the amplification matrix $\mathbf{A}$ [6,6] and the load vector $\mathbf{q}_n$ [6] which is independent of $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$. It is obvious that the [6,6] amplification matrix of the generalized scheme consists of four unnecessary roots compared to the exact amplification matrix in (8). Convergency requires two complex conjugate roots with magnitude one, all other roots must be smaller than one. Katona and Zienkiewicz [10] for example fit their free parameters to get the desired spectral properties. In the following a new principle is introduced to get a two or three root scheme from (25) in a more straight forward sense. The idea is to use some of the following reasonable equations and incorporate them into the scheme (21) - (24) :

$$\dot{\mathbf{d}}_n = \mathbf{v}_n \tag{26}$$

$$\ddot{\mathbf{d}}_n = \dot{\mathbf{v}}_n \tag{27}$$

$$\dot{\mathbf{v}}_n = \mathbf{M}^{-1}\,[\,\mathbf{p}_n - \mathbf{C}\,\mathbf{v}_n - \mathbf{K}\,\mathbf{d}_n\,] \tag{28}$$

Inserting the equations (26) - (28) in the equations (21) - (24) we get a generalized three root scheme. The next step would be to also remove the third root in a direct way by using the requirement,

$$\ddot{\mathbf{v}}_n = \mathbf{M}^{-1}\,[\,\dot{\mathbf{p}}_n - \mathbf{C}\,\dot{\mathbf{v}}_n - \mathbf{K}\,\mathbf{v}_n\,] \tag{29}$$

This implies in the nonlinear case (1) to calculate the time derivative of the internal forces,

$$\dot{\mathbf{f}}_n = \mathbf{K}_n\,\mathbf{v}_n, \qquad \mathbf{K}_n = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_n \tag{30}$$

If we focus our interest on explicit schemes, it is not desired to calculate the tangent stiffness matrix $\mathbf{K}_n$ because of the additional numerical effort. To avoid this the third root

may be removed in the conventional way through the spectral properties by fitting the free parameters or it may be used to introduce numerical dissipation. At the beginning of the time interval the state vectors $\mathbf{d}_n$, $\mathbf{v}_n$, $\dot{\mathbf{v}}_n$ and $\ddot{\mathbf{v}}_n$ are known. The new state at the end of the time interval is determined through the equations (21) - (24) together with (26) - (28). The parameter Π_1 drops out through equation (27). By setting $\Psi_3 = 0$ the algorithm becomes explicit :

$$\begin{aligned} \kappa_2 \Psi_2 \mathbf{M} \Delta\ddot{\mathbf{v}} \Delta t \;=\; & \mathbf{p}_n \;+\; \Psi_1 \dot{\mathbf{p}}_n \Delta t \;+\; \frac{1}{2} \Psi_2 \ddot{\mathbf{p}}_n \Delta t^2 \\ & - \mathbf{M} \left(\dot{\mathbf{v}}_n + \Psi_1 \ddot{\mathbf{v}}_n \Delta t \right) \\ & - \mathbf{C} \left(\mathbf{v}_n + \Psi_1 \dot{\mathbf{v}}_n \Delta t + \eta_2 \Psi_2 \ddot{\mathbf{v}}_n \Delta t^2 \right) \\ & - \mathbf{K} \left(\mathbf{d}_n + \Psi_1 \mathbf{v}_n \Delta t + \beta_2 \Psi_2 \dot{\mathbf{v}}_n \Delta t^2 \right) \end{aligned} \tag{31}$$

$$\gamma_2 \Pi_2 \Delta\ddot{\mathbf{d}} \Delta t \;=\; \eta_2 \Pi_2 \ddot{\mathbf{v}}_n \Delta t^2 + \eta_3 \Pi_3 \Delta\ddot{\mathbf{v}} \Delta t^2 \tag{32}$$

$$\begin{aligned} \mathbf{d}_{n+1} &= \mathbf{d}_n + \mathbf{v}_n \Delta t + \beta_2 \dot{\mathbf{v}}_n \Delta t^2 + \beta_3 \Delta\ddot{\mathbf{d}} \Delta t^2 \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \dot{\mathbf{v}}_n \Delta t + \eta_2 \ddot{\mathbf{v}}_n \Delta t^2 + \eta_3 \Delta\ddot{\mathbf{v}} \Delta t^2 \\ \ddot{\mathbf{v}}_{n+1} &= \ddot{\mathbf{v}}_n + \Delta\ddot{\mathbf{v}} \end{aligned} \tag{33}$$

$$\dot{\mathbf{v}}_{n+1} = \mathbf{M}^{-1} \left(\mathbf{p}_{n+1} - \mathbf{C}\, \mathbf{v}_{n+1} - \mathbf{K}\, \mathbf{d}_{n+1} \right) \tag{34}$$

So far the algorithm (31)- (34) contains some parameters which are redundant. After stability and accuracy investigations a final version in the form of a predictor corrector scheme will be presented.

4. CONVERGENCE INVESTIGATIONS

4.1 Amplification Matrix

For simplicity the theoretical considerations of stability and accuracy start with the linear undamped single degree of freedom system

$$\ddot{u}(t) + \omega^2 u(t) = 0 \tag{35}$$

It will be shown later under which circumstances the following proofs of accuracy and stability hold true for multidegree of freedom systems with nonlinearites. Physical damping is not considered here because it is less important in explicit schemes which are mainly applied to short duration problems.

For the following theoretical considerations the scheme (31)- (34) must be transformed into the form (25). The state vector consists of three members

$$x_n^T = [d_n,\ \Delta t\, v_n,\ \Delta t^3\, \ddot{v}_n] \tag{36}$$

After some transformations of (31)- (34) we get the amplification matrix **A** defined by equation (25):

$$\mathbf{A} = [a_{ij}] \tag{37}$$

$$a_{11} = 1 - \beta_2 \Omega^2 + \beta_2 \frac{\eta_3 \Pi_3}{\gamma_2 \Pi_2} \frac{\beta_3}{\kappa_2} \Omega^4$$

$$a_{12} = 1 - \beta_3 \frac{1}{\kappa_2 \Psi_2} \frac{\eta_3 \Pi_3}{\gamma_2 \Pi_2} \Omega^2$$

$$a_{13} = \beta_3 \left(\frac{\eta_2}{\gamma_2} - \frac{1}{\kappa_2 \Psi_2} \frac{\eta_3 \Pi_3}{\gamma_2 \Pi_2} \Psi_1 \right)$$

$$a_{21} = -\Omega^2 + \eta_3 \frac{\beta_2}{\kappa_2} \Omega^4$$

$$a_{22} = 1 - \frac{1}{\kappa_2 \Psi_2} \eta_3 \Psi_1 \Omega^2$$

$$a_{23} = \eta_2 - \frac{1}{\kappa_2 \Psi_2} \eta_3 \Psi_1$$

$$a_{31} = \frac{\beta_2}{\kappa_2} \Omega^4$$

$$a_{32} = -\frac{1}{\kappa_2 \Psi_2} \Psi_1 \Omega^2$$

$$a_{33} = 1 - \frac{1}{\kappa_2 \Psi_2} \Psi_1$$

where Ω denotes the dimensionless frequency

$$\Omega = \omega \Delta t \tag{38}$$

The amplification matrix possesses the three invariants

$$\begin{aligned} I_A &= (a_{11} + a_{22} + a_{33}) \\ II_A &= a_{11} a_{22} + a_{22} a_{33} + a_{33} a_{11} - a_{12} a_{21} - a_{13} a_{31} - a_{23} a_{32} \\ III_A &= \det \mathbf{A} \end{aligned} \tag{39}$$

4.2 Accuracy

To transform (25) to the corresponding difference equation in the displacements only we write (25) for three adjacent time steps n+k, k=1,2,3 respectively. From these 9 equations we are able to eliminate the 8 values v_{n+i} and $\dot{v}_{n+i}$ with i = 0,1...3 to get a difference equation in the displacements d_{n+i} :

$$d_{n+3} - I_A d_{n+2} + II_A d_{n+1} - III_A d_n = 0 \tag{40}$$

The accuracy of an algorithm is commonly defined by replacing the numerical solution d_{n+i} through the exact solution $u(t+i\Delta t)$ in (40) and considering the residual

$$R = u(t+3\Delta t) - I_A u(t+2\Delta t) + II_A u(t+\Delta t) - III_A u(t) = O(\Delta t^{p+2}) \tag{41}$$

where p denotes the order of accuracy, see the standard textbooks [3] or [34] for the definitions. If we develop the exact solution $u(t+i\Delta t)$ in Taylor series around t and substitute the differential equation (35) we get the order p of the algorithm which must be at least $p \geq 1$ to be consistent. If we require a certain accuracy p we have to cancel the terms up to $O(\Delta t^{p+1})$ in (41) by adjusting the free parameters in the following manner:

$$\beta_2 = 1 - \eta_2 \qquad \text{for } p=2 \tag{42}$$

$$\frac{\beta_3}{\gamma_2} = \frac{\eta_3 - \frac{1}{12} - \eta_2 (1-\eta_2) (1-\frac{1}{\kappa_2}) - \eta_2}{\eta_3 \frac{\Pi_3}{\Pi_2} (1-\frac{1}{\Psi_1}) - \eta_2} \qquad \text{for } p=3 \tag{43}$$

$$\frac{\Pi_3}{\Pi_2} = \frac{1}{\eta_3 \frac{\beta_3}{\gamma_2}(2-\frac{1}{\Psi_1})} \left\{ \frac{\beta_3}{\gamma_2}\eta_2 \left[2-\frac{1}{\kappa_2}(1-\eta_2) \right] + \eta_3 (2-\eta_2) \right. \tag{44}$$

$$\left. -\eta_2 (3-2\eta_2) - (1-\eta_2)\frac{1}{\kappa_2}(\eta_3 - 2\eta_2) \right\} \quad \text{for } p=4$$

4.3 Stability

Stability which is investigated using the three eigenvalues of the amplification matrix **A** is called spectral stability, see [3]. The eigenvalues of the matrix **A** govern the long time behavior of the algorithm. The solution after repeated use of (25) omitting the load term is:

$$\mathbf{x}_n = \mathbf{A}^n \mathbf{x}_0 \tag{45}$$

The unsymmetric matrix **A** can be decomposed in its left hand side eigenvectors **Y** and its eigenvalues **N** :

$$\mathbf{A} = \mathbf{Y}^{-1} \mathbf{N} \mathbf{Y} \tag{46}$$

With (46) the numerical solution from (45) can be expressed in the following inequality with 2- norms :

$$||\mathbf{x}_n|| \leq ||\mathbf{Y}^{-1}||\ ||\mathbf{N}||^n\ ||\mathbf{Y}||\ ||\mathbf{x}_0|| \tag{47}$$

The numerical algorithm is called stable if the solution of the homogeneous problem does not grow $||\mathbf{x}_n|| \leq ||\mathbf{x}_0||$. For $n \gg 1$ the term $||\mathbf{N}||^n$ in (47) becomes dominant and it is sufficient to require

$$||\mathbf{N}|| \leq 1 \tag{48}$$

with

$$\mathbf{N} = \text{diag}(\nu_i) \quad i=1,2,3$$

From (48) follows that the eigenvalues of **A** have to remain in or on the unit circle of the complex plane

$$\rho \leq 1 \tag{49}$$

$$\rho = \max \left\{ |\nu_i| \right\} \quad i=1,2,3$$

where ρ is called the spectral radius of **A**. The eigenvalues of **A** can be determined from the characteristic polynomial

$$-\det(\mathbf{A} - \nu \mathbf{I}) = \nu^3 - I_A \nu^2 + II_A \nu - III_A = 0 \tag{50}$$

The stability investigations follow standard procedures, for details see [14], [3] and [34]. The free parameters are adjusted to get a maximum in the stability limit. It turns out that the stability limit decreases drastically if the parameter Ψ_1 is not set to unity i.e., we require

$$\Psi_1 = 1 \tag{51}$$

The limit case $\Omega \to 0$ is important for the convergence with small time steps $\Delta t \to 0$. A spectral radius $\rho = 1$ and a vanishing spurious root $\nu_3 = 0$ for $\Omega \to 0$ is insured by

$$\frac{\Psi_1}{\Psi_2 \kappa_2} = 1 \tag{52}$$

After substituting (42), (51), (52) in the algorithm (31)- (34) the parameters Π_2 and γ_2 come out to be redundant. Therefore they are set to

$$\Pi_2 = \gamma_2 = 1 \tag{53}$$

4.4 Convergence for Linear Multidegree of Freedom Systems

So far the accuracy and stability proofs only are valid for single degree of freedom systems. It is well known from linear multistep methods of the form (9) that the theory can be extended to multidegree of freedom systems as long as the system (4) can be decoupled into N scalar equations, see the standard references [1], [3], [34]. This supposes a modal damping matrix composed of linear combinations of mass and stiffness matrices. The scheme presented shows a nonlinear architecture, see formula (10). It must be demonstrated that the nonlinear scheme can be decoupled to justify the foregoing convergence proof.

To reduce the number of free parameters the equations (42) and (51)- (53) are substituted into the scheme (31)- (34). As in the single degree of freedom case the scheme (31)- (34) can also be transformed to a form like (25) for multidegree of freedom systems :

$$
\begin{aligned}
\mathbf{d}_{n+1} &= [\, \mathbf{I} - (1-\eta_2)\,\Delta t^2\,\mathbf{M}^{-1}\mathbf{K} + \beta_3\,\eta_3\,\Pi_3\,\Delta t^3\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{K} \\
&\quad + (1-\eta_2)\,\Psi_2\,\beta_3\,\eta_3\,\Pi_3\,\Delta t^4\,\mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{K}\,]\;\mathbf{d}_n \\
&+ [\, \mathbf{I} - (1-\eta_2)\,\Delta t\,\mathbf{M}^{-1}\mathbf{C} - \beta_3\,\eta_3\,\Pi_3\,\Delta t^2\,\mathbf{M}^{-1}\mathbf{K} \\
&\quad + \beta_3\,\eta_3\,\Pi_3\,\Delta t^2\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{C} + (1-\eta_2)\,\Psi_2\,\beta_3\,\eta_3\,\Pi_3\,\Delta t^3\,\mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{C}\,]\;\Delta t\,\mathbf{v}_n \\
&+ [\, \beta_3\,(\,1-\eta_3\,\Pi_3\,)\,\mathbf{I} - \eta_2\,\Psi_2\,\beta_3\,\eta_3\,\Pi_3\,\Delta t\mathbf{M}^{-1}\mathbf{C}\,]\;\Delta t^3\,\ddot{\mathbf{v}}_n + \mathbf{q}_{dn} \\
\Delta t\,\mathbf{v}_{n+1} &= [\, -\Delta t^2\,\mathbf{M}^{-1}\mathbf{K} + \eta_3\,\Delta t^3\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{K} + (1-\eta_2)\,\Psi_2\,\eta_3\,\Delta t^4\,\mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{K}\,]\;\mathbf{d}_n \\
&+ [\, \mathbf{I} - \Delta t\,\mathbf{M}^{-1}\mathbf{C} - \eta_3\,\Delta t^2\,\mathbf{M}^{-1}\mathbf{K} \qquad (54) \\
&\quad + \eta_3\,\Delta t^2\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{C} + (1-\eta_2)\,\Psi_2\,\eta_3\,\Delta t^3\,\mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{C}\,]\;\Delta t\,\mathbf{v}_n \\
&+ [\, (\,\eta_2 - \eta_3\,)\,\mathbf{I} - \eta_2\,\Psi_2\,\eta_3\,\Delta t\,\mathbf{M}^{-1}\mathbf{C}\,]\;\Delta t^3\,\ddot{\mathbf{v}}_n + \mathbf{q}_{vn} \\
\Delta t^3\,\ddot{\mathbf{v}}_{n+1} &= [\, \Delta t^3\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{K} + \beta_2\,\Psi_2\,\Delta t^4\,\mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{K}\,]\;\mathbf{d}_n \\
&+ [\, -\Delta t^2\,\mathbf{M}^{-1}\mathbf{K} + \Delta t^2\,\mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{C} + \beta_2\,\Psi_2\,\Delta t^3\,\mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{C}\,]\;\Delta t\mathbf{v}_n \\
&- \eta_2\,\Psi_2\,\Delta t\,\mathbf{M}^{-1}\mathbf{C}\,\Delta t^3\,\ddot{\mathbf{v}}_n + \mathbf{q}_{\dot{v}n}
\end{aligned}
$$

The load vectors $\mathbf{q}_{dn}$, $\mathbf{q}_{vn}$, $\mathbf{q}_{\dot{v}n}$ are independent of the state n and n+1. To decouple the equation (54) we use the eigenvalue problem corresponding to the undamped system (4).

$$\mathbf{K}\,\Phi = \mathbf{M}\,\Phi\,\Lambda \qquad (55)$$

with the eigenvector matrix Φ and the diagonal eigenvalue matrix $\Lambda = \mathrm{diag}\,(\,\omega_i^2\,)$; i = 1,...N. The eigenvectors fulfill the orthogonality conditions

$$
\begin{aligned}
\mathbf{I} &= \Phi^T\,\mathbf{M}\,\Phi \qquad (56) \\
\Lambda &= \Phi^T\,\mathbf{K}\,\Phi
\end{aligned}
$$

If we assume a damping matrix which can be diagonalized through the eigenvectors Φ :

$$\Xi = \Phi^T\,\mathbf{C}\,\Phi = \mathrm{diag}\,(\,2\,\xi_i\,\omega_i\,) \qquad (57)$$

the problem (4) can be completely decoupled with the transformation

$$\mathbf{u} = \Phi\,\mathbf{u}^g \qquad (58)$$

If we premultiply (4) with $\Phi^T\,\mathbf{M}$, insert (58) and use the condition (56) and (57) the decoupled equation comes out

$$\mathbf{I}\,\ddot{\mathbf{u}}^g + \Xi\,\dot{\mathbf{u}}^g + \Lambda\,\mathbf{u}^g = \mathbf{p}^g \qquad (59)$$

In the formulas (54) mixed matrix products of the stiffness, mass and damping occur. From (56) and (57) we get

$$\begin{aligned} \mathbf{M}^{-1} &= \Phi\,\Phi^{T} \\ \mathbf{K} &= \Phi^{-T}\,\Lambda\,\Phi^{T}\,\mathbf{M} \\ \mathbf{C} &= \Phi^{-T}\,\Xi\,\Phi^{T}\,\mathbf{M} \end{aligned} \tag{60}$$

With (60) we get the following expressions for the mixed matrix products in (54)

$$\begin{aligned} \mathbf{M}^{-1}\mathbf{K} &= \Phi\,\Lambda\,\Phi^{T}\,\mathbf{M} \\ \mathbf{M}^{-1}\mathbf{C} &= \Phi\,\Xi\,\Phi^{T}\,\mathbf{M} \\ \mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{K} &= \Phi\,\Lambda^{2}\,\Phi^{T}\,\mathbf{M} \\ \mathbf{M}^{-1}\mathbf{K}\,\mathbf{M}^{-1}\mathbf{C} &= \Phi\,\Lambda\,\Xi\,\Phi^{T}\,\mathbf{M} \\ \mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{K} &= \Phi\,\Xi\,\Lambda\,\Phi^{T}\,\mathbf{M} \\ \mathbf{M}^{-1}\mathbf{C}\,\mathbf{M}^{-1}\mathbf{C} &= \Phi\,\Xi^{2}\,\Phi^{T}\,\mathbf{M} \end{aligned} \tag{61}$$

If we transform all state vectors (54) with (58) and premultiply with $\Phi^{T}\,\mathbf{M}$ all equations are decoupled using the relations (56a) and (61). As an example we get for the displacements

$$\mathbf{d}_{n+1} = \Phi\,\mathbf{d}^{g}_{n+1} \tag{62}$$

$$\begin{aligned} \mathbf{d}^{g}_{n+1} = {} & [\,\mathbf{I} - (1-\eta_2)\,\Delta t^2\,\Lambda + \beta_3\,\eta_3\,\Pi_3\;\Delta t^3\,\Xi\,\Lambda \\ & + (1-\eta_2)\,\Psi_2\,\beta_3\,\eta_3\,\Pi_3\;\Delta t^4\,\Lambda^2\,]\,\mathbf{d}^{g}_{n} \\ & + [\,\mathbf{I} - (1-\eta_2)\,\Delta t\Xi - \beta_3\,\eta_3\,\Pi_3\,\Delta t^2\,\Lambda \\ & + \beta_3\,\eta_3\,\Pi_3\;\Delta t^2\,\Xi^2 + (1-\eta_2)\,\Psi_2\,\beta_3\,\eta_3\,\Pi_3\;\Delta t^3\,\Lambda\,\Xi\,]\,\Delta t\,\mathbf{v}^{g}_{n} \\ & + [\,\beta_3\,(\,1-\eta_3\,\Pi_3\,)\,\mathbf{I} - \eta_2\,\Psi_2\,\beta_3\,\eta_3\,\Pi_3\,\Delta t\,\Xi\,]\,\Delta t^3\,\ddot{\mathbf{v}}^{g}_{n} + \mathbf{q}^{g}_{dn} \end{aligned} \tag{63}$$

The equations (54b,c) for $\mathbf{v}^{g}_{n+1}$ and $\ddot{\mathbf{v}}^{g}_{n+1}$ can be decoupled in the same way. Only diagonal matrices appear in (63). In spite of the nonlinear architecture the scheme (31)- (34) can be decoupled and the foregoing convergence proof for single degree of freedom systems holds also for multidegree of freedom systems. If the damping matrix **C** can not be decoupled through the undamped eigenvectors the whole convergence proof fails for both types of algorithms (9) and (10). For an arbitrary type of damping see the work of Thomas and Gladwell [35] and Gear [36].

4.5 Convergence for Nonlinear Systems

To study convergence for the nonlinear case (1) damping effects are neglected. The equations (42) and (51)- (53) are substituted in (31)- (34). After some transformations we get

$$\begin{aligned}
\mathbf{d}_{n+1} &= \mathbf{d}_n + \Delta t\, \mathbf{v}_n + (1 - \eta_3 \Pi_3)\, \ddot{\mathbf{v}}_n \Delta t^3 \\
&\quad - (1 - \eta_2 - \beta_3 \eta_3 \Pi_3)\, \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}_n) - \beta_3 \eta_3 \Pi_3 \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}^0_{n+1}) + \mathbf{q}_{dn} \\
\Delta t\, \mathbf{v}_{n+1} &= \Delta t\, \mathbf{v}_n + (\eta_2 - \eta_3)\, \ddot{\mathbf{v}}_n \Delta t^3 \\
&\quad - (1 - \eta_3)\, \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}_n) - \eta_3 \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}^0_{n+1}) + \mathbf{q}_{vn} \\
\Delta t^3\, \ddot{\mathbf{v}}_{n+1} &= + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}_n) - \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}^0_{n+1}) + \mathbf{q}_{\dot{v}n}
\end{aligned} \tag{64}$$

with

$$\mathbf{d}^0_{n+1} = \mathbf{d}_n + \Delta t\, \mathbf{v}_n - (1 - \eta_2)\, \Psi_2\, \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}_n) + \mathbf{q}_{d0}$$

For the convergence proof we need a difference equation in the displacements similar to equation (40) for the linear system. The velocities $\mathbf{v}_n$ and the derivatives of the acceleration $\ddot{\mathbf{v}}_n$ must be eliminated through additional time steps n, n-1. After some transformations the following difference equation results :

$$\begin{aligned}
\mathbf{d}_{n+1} - 2\,\mathbf{d}_n + \mathbf{d}_{n-1} &= \mathbf{q}_d - [\, \eta_3 (\Pi_3 - 1)]\, \Delta t^2 \mathbf{M}^{-1} [\, \mathbf{f}(\mathbf{d}^0_{n-1}) - \mathbf{f}(\mathbf{d}_{n-2})] \\
&\quad - \eta_2\, \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}_{n-1}) \\
&\quad - [\, \eta_2 - \eta_3 (\Pi_3 - \eta_3) - \beta_3 \eta_3 \Pi_3]\, \Delta t^2 \mathbf{M}^{-1} [\, \mathbf{f}(\mathbf{d}^0_n) - \mathbf{f}(\mathbf{d}_{n-1})] \\
&\quad - (1 - \eta_2)\, \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{d}_n) \\
&\quad - \beta_3 \eta_3 \Pi_3 \Delta t^2 \mathbf{M}^{-1} [\, \mathbf{f}(\mathbf{d}^0_{n+1}) - \mathbf{f}(\mathbf{d}_n)]
\end{aligned} \tag{65}$$

As in the linear case the local truncation error is commonly defined from the homogeneous part of (65). If we replace the numerical solution **d** by the exact solution **u** the residual of equation (65) provides the order of accuracy

$$\begin{aligned} \mathbf{r} \;=\; & \mathbf{u}_{n+1} - 2\,\mathbf{u}_n + \mathbf{u}_{n-1} \\ & + [\,\eta_3\,(\,\Pi_3 - 1\,)\,]\,\Delta t^2\,\mathbf{M}^{-1}\,[\,\mathbf{f}(\mathbf{u}^0_{n-1}) - \mathbf{f}(\mathbf{u}_{n-2})\,] + \eta_2\,\Delta t^2\,\mathbf{M}^{-1}\,\mathbf{f}\,(\mathbf{u}_{n-1}) \\ & + [\,\eta_2 - \eta_3(\,\Pi_3 - \eta_3) - \beta_3\,\eta_3\,\Pi_3\,]\,\Delta t^2\,\mathbf{M}^{-1}\,[\,\mathbf{f}(\mathbf{u}^0_n) - \mathbf{f}(\mathbf{u}_{n-1})\,] \\ & + (\,1 - \eta_2\,)\,\Delta t^2\,\mathbf{M}^{-1}\,\mathbf{f}\,(\mathbf{u}_n) + \beta_3\,\eta_3\,\Pi_3\,\Delta t^2\,\mathbf{M}^{-1}\,[\,\mathbf{f}(\mathbf{u}^0_{n+1}) - \mathbf{f}(\mathbf{u}_n)\,] \\ = \; & \mathbf{g}\,(\,\mathbf{u},\,\mathbf{M}^{-1}\mathbf{f}(\mathbf{u})\,)\;O(\Delta t^{\,p+2}) \end{aligned} \tag{66}$$

with the convergence rate p. The nonlinear internal forces $\mathbf{f}(\mathbf{u})$ must fulfill the Lipschitz continuity in the time interval $t_{n-1} \le t \le t_{n+1}$

$$|\,|\,\mathbf{f}(\bar{\mathbf{u}}(t)) - \mathbf{f}(\mathbf{u}(t))\,|\,| \;\le\; L\,|\,|\,\bar{\mathbf{u}}(t) - \mathbf{u}(t)\,|\,| \tag{67}$$

To get the residual $\mathbf{r}$ the displacements $\mathbf{u}$ and the internal forces $\mathbf{f}(\mathbf{u})$ are developed in Taylor series around the point n and the differential equation (1) is substituted several times.

For $\eta_2 = \eta_3 = \beta_3 = 0$ and $\Psi_2 = \Pi_3 = 1$ we obtain the central difference method from (65) :

$$\mathbf{d}_{n+1} - 2\,\mathbf{d}_n + \mathbf{d}_{n-1} \;=\; \mathbf{q}_d - \Delta t^2\,\mathbf{M}^{-1}\,\mathbf{f}\,(\mathbf{d}_n) \tag{68}$$

with the truncation error from (66)

$$\mathbf{r} \;=\; \mathbf{g}\,(\,\mathbf{u},\,\mathbf{M}^{-1}\mathbf{f}(\mathbf{u})\,)\;\frac{1}{24}\,\Delta t^4 \tag{69}$$

The central difference method possesses a rate of convergence p=2 for arbitrary Lipschitz continuous internal forces, which is well known.

For the general algorithm (65) higher accuracy was proved for linear internal forces in the preceding section. In the worst possible case of strong nonlinearities the general algorithm (65) may decrease to second order accuracy p=2. For smooth and weak nonlinearities the higher order of the linear case will be maintained.

To investigate the stability of the scheme (65) we must define the coefficients of the left hand side of equation (65) by

$$\mathbf{d}_{n+1} - 2\,\mathbf{d}_n + \mathbf{d}_{n-1} \;:=\; \sum_{i=0}^{k} \alpha_i\,\mathbf{d}_{n-1+i} \tag{70}$$

The corresponding characteristic polynomial

$$\sigma(\zeta) \;=\; \sum_{i=0}^{k} \alpha_i\,\zeta^i \;=\; 1 - 2\,\zeta + \zeta^2 \tag{71}$$

governs the stability behavior for the limit case $n \to \infty$ and $\Delta t \to 0$. If the roots of (71) are within or on the unit circle $|\zeta_i| \leq 1$ the method is stable, see the definition in [3], chapter III 10. This kind of stability is a necessary condition for the spectral stability (49). Since the roots of (71) are $\zeta_{1,2} = 1$ the scheme (65) is stable for small time steps and for nonlinear forces which fulfill the Lipschitz condition (67).

The stability boundaries (critical time steps) which were already provided for the linearized system must be used carefully in the nonlinear system. To determine the sources of errors in the linearization it is necessary to evaluate the internal forces in Taylor series around $\mathbf{u}_n$ with the incrementation $\mathbf{u}(t_n+\tau) = \mathbf{u}_n + \Delta\mathbf{u}(\tau)$

$$\mathbf{f}(\mathbf{u}_n + \Delta\mathbf{u}) = \mathbf{f}(\mathbf{u}_n) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{u_n} \Delta\mathbf{u} + \frac{1}{2} \left.\frac{\partial^2 \mathbf{f}}{\partial \mathbf{u}\, \partial \mathbf{u}}\right|_{u_n} \Delta\mathbf{u}\, \Delta\mathbf{u} + \cdots \tag{72}$$

As long as the third term in (72) is sufficiently small the nonlinear differential equation (1) can be considered as a linear equation in the time interval $t_n \leq t \leq t_{n+1}$ with $\tau = t - t_n$

$$\mathbf{M}\, \ddot{\mathbf{u}}(\tau) + \mathbf{K}_n\, \Delta\mathbf{u}(\tau) = \mathbf{p}(\tau) - \mathbf{f}(\mathbf{u}_n) \tag{73}$$

$$\text{with} \quad \mathbf{K}_n = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{u_n}$$

and the critical time steps of the linear case can then be used with respect to the actual tangent stiffness $\mathbf{K}_n$. The linearization needs of course Lipschitz continuous internal forces $\mathbf{f}(\mathbf{u})$ so that the stiffness is bounded. In addition the second order term in (72) must remain small

$$\left| \left.\frac{\partial^2 \mathbf{f}}{\partial \mathbf{u}\, \partial \mathbf{u}}\right|_{u_n} \Delta\mathbf{u}\, \Delta\mathbf{u} \right| \leq \epsilon \left| \mathbf{f}(\mathbf{u}_n) \right| \tag{74}$$

$$\left| \left.\frac{\partial^2 f_i}{\partial u_j\, \partial u_k}\right|_{u_n} \Delta u_j\, \Delta u_k \right| \leq \epsilon \left| f_i(\mathbf{u}_n) \right|$$

with $\epsilon \ll 1$. In (74b) the sum must be taken over repeated indices. If (74) is not satisfied the derived critical time steps of the linearized system are meaningless and the algorithm shows unstable results. A large sudden stiffening $\Delta\mathbf{K}$ causes an increase of the quantity (74), see example 6.2. Furthermore large displacement increments $\Delta\mathbf{u}$ can lead to a large quantity (74) even for small stiffness changes. This happens in softening systems, see the examples 6.3, 6.4, 6.5. One might expect that softening systems allow a larger critical

time step. But the term (74) shows clearly that even with small time steps large displacement increments Δu can destroy the stability.

As a consequence of these stability considerations one must use time steps smaller than the critical time step derived from the linearized system as well as incorporate equilibrium, displacement and energy control mechanisms to insure the stability of the algorithm for an arbitrary nonlinear system.

5. GENERALIZED EXPLICIT ALGORITHM FOR NONLINEAR PROBLEMS

5.1 Final Algorithm

The algorithm (31)- (34) is applied to the general nonlinear problem (1). The parameters Ψ_1, κ_2, Π_2, γ_2, are eliminated with (42) and (51)- (53). Five parameters remain. A particularly defined acceleration $\dot{\mathbf{v}}^{*}_{n+1}$ as the primary unknown minimizes the computational effort :

$$\dot{\mathbf{v}}^{*}_{n+1} = \dot{\mathbf{v}}_n + \ddot{\mathbf{v}}_n \Delta t + \Delta\ddot{\mathbf{v}} \Delta t \tag{75}$$

The unknown vectors $\Delta\ddot{\mathbf{v}}$ and $\Delta\ddot{\mathbf{d}}$ are eliminated from (33) and (34) with (31) and (32). We get after some transformations :

$$\begin{aligned} \mathbf{M}\,\dot{\mathbf{v}}^{*}_{n+1} &= \mathbf{p}_n + \dot{\mathbf{p}}_n \Delta t + \frac{1}{2}\Psi_2 \ddot{\mathbf{p}}_n \Delta t^2 \\ &\quad - \mathbf{f}\left(\mathbf{d}_n + \mathbf{v}_n \Delta t + (1-\eta_2)\Psi_2 \dot{\mathbf{v}}_n \Delta t^2 ,\ \mathbf{v}_n + \dot{\mathbf{v}}_n \Delta t + \eta_2 \Psi_2 \ddot{\mathbf{v}}_n \Delta t^2 \right) \end{aligned} \tag{76}$$

$$\begin{aligned} \mathbf{d}_{n+1} &= \mathbf{d}_n + \mathbf{v}_n \Delta t + (1 - \eta_2 - \beta_3 \eta_3 \Pi_3)\,\dot{\mathbf{v}}_n \Delta t^2 \\ &\quad + \beta_3 (\eta_2 - \eta_3 \Pi_3)\,\ddot{\mathbf{v}}_n \Delta t^3 + \beta_3 \eta_3 \Pi_3 \dot{\mathbf{v}}^{*}_{n+1} \Delta t^2 \end{aligned}$$

$$\begin{aligned} \mathbf{v}_{n+1} &= \mathbf{v}_n + (1 - \eta_3)\,\dot{\mathbf{v}}_n \Delta t \\ &\quad + (\eta_2 - \eta_3)\,\ddot{\mathbf{v}}_n \Delta t^2 + \eta_3 \dot{\mathbf{v}}^{*}_{n+1} \Delta t \end{aligned} \tag{77}$$

$$\ddot{\mathbf{v}}_{n+1} = (\dot{\mathbf{v}}^{*}_{n+1} - \dot{\mathbf{v}}_n)\,\frac{1}{\Delta t}$$

$$\dot{\mathbf{v}}_{n+1} = \mathbf{M}^{-1}\left(\mathbf{p}_{n+1} - \mathbf{f}\left(\mathbf{d}_{n+1}, \mathbf{v}_{n+1} \right) \right) \tag{78}$$

The algorithm (76)- (78) can be presented in a final predictor corrector scheme. Before

starting the time step loop consistent starting conditions must be calculated from given $\mathbf{d}(0) = \mathbf{d}_0$ and $\mathbf{v}(0) = \mathbf{v}_0$

$$\dot{\mathbf{v}}_0 = \mathbf{M}^{-1} (\mathbf{p}_0 - \mathbf{f}_0) \tag{79}$$

$$\ddot{\mathbf{v}}_0 = \mathbf{M}^{-1} (\dot{\mathbf{p}}_0 - \mathbf{K}_0 \mathbf{v}_0) \qquad \text{with} \quad \mathbf{K}_0 = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{u_0}$$

The following steps must be carried out in each time interval :

1. Start at time t_n with the known quantities $\mathbf{d}_n$, $\mathbf{v}_n$, $\dot{\mathbf{v}}_n$, and $\ddot{\mathbf{v}}_n$
2. Calculate the predictor values

$$\mathbf{d}^*_{n+1} = \mathbf{d}_n + \mathbf{v}_n \Delta t + (1 - \eta_2) \Psi_2 \dot{\mathbf{v}}_n \Delta t^2 \tag{80}$$

$$\mathbf{v}^*_{n+1} = \mathbf{v}_n + \dot{\mathbf{v}}_n \Delta t + \eta_2 \Psi_2 \ddot{\mathbf{v}}_n \Delta t^2$$

$$\mathbf{f}^*_{n+1} = \mathbf{f} (\mathbf{d}^*_{n+1} , \mathbf{v}^*_{n+1})$$

$$\mathbf{p}^*_{n+1} = \mathbf{p}_n + \dot{\mathbf{p}}_n \Delta t + \frac{1}{2} \Psi_2 \ddot{\mathbf{p}}_n \Delta t^2$$

3. Get the accelerations from the equilibrium equations

$$\dot{\mathbf{v}}^*_{n+1} = \mathbf{M}^{-1} [\mathbf{p}^*_{n+1} - \mathbf{f}^*_{n+1}] \tag{81}$$

4. Calculate the corrector values of the state vectors :

$$\ddot{\mathbf{v}}_{n+1} = (\dot{\mathbf{v}}^*_{n+1} - \dot{\mathbf{v}}_n) \frac{1}{\Delta t} \tag{82}$$

$$\mathbf{v}_{n+1} = \mathbf{v}^*_{n+1} + [\eta_2 (1 - \Psi_2) - \eta_3] \ddot{\mathbf{v}}_n \Delta t^2 + \eta_3 \ddot{\mathbf{v}}_{n+1} \Delta t^2$$

$$\mathbf{d}_{n+1} = \mathbf{d}^*_{n+1} + (1-\eta_2) (1-\Psi_2) \dot{\mathbf{v}}_n \Delta t^2 + \beta_3 (\eta_2 - \eta_3 \Pi_3) \ddot{\mathbf{v}}_n \Delta t^3 + \beta_3 \eta_3 \Pi_3 \ddot{\mathbf{v}}_{n+1} \Delta t^3$$

5. Calculate the final values for the accelerations and the internal forces at the end of the time step :

$$\mathbf{f}_{n+1} = \mathbf{f} (\mathbf{d}_{n+1} , \mathbf{v}_{n+1}) \tag{83}$$

$$\dot{\mathbf{v}}_{n+1} = \mathbf{M}^{-1} (\mathbf{p}_{n+1} - \mathbf{f}_{n+1})$$

The explicit algorithm is at least second order accurate. To increase the accuracy as high as possible and get a stability limit as large as possible the remaining five parameters are adjusted with respect to the equations (43), (44) and the stability condition (49). The results are presented in table 1 and 2 where a $\hat{}$ is attached to the five parameters to distinguish between the generalized algorithm and the individual methods.

The generalized algorithm includes known methods in the sense of spectral identity and some new useful higher order versions. The well known central difference method and the Newmark family recently developed by Kujawski and Gallagher [8] can be produced in a one step form. Furthermore a fourth order algorithm could be developed which possesses mildly numerical damping controllable by the parameter η_3. If high numerical damping is desired a third order algorithm can be used which has the important advantage of a stability limit which is very close to the central difference method. The increase in accuracy costs only about 3 % more computational effort. Further trials to enlarge the stability limit failed.

In general the algorithm needs two function evaluation per time step, see equation (80c) and (83a). The central difference method for example needs only one function evaluation per time step. This leads to the requirement that the stability limit of the new algorithm must be nearly two times larger than the stability limit of the central difference method to be competitive with respect to the computational effort. The cost of additional auxiliary vectors can be neglected compared to the cost of function evaluations.

5.2 Remarks on Possible Multicorrector Steps

The scheme appears to offer the possibility of multi corrector steps within (81) - (83). Start with $\dot{\mathbf{v}}_{n+1}^{(1)} = \dot{\mathbf{v}}_{n+1}$, $\mathbf{v}_{n+1}^{(1)} = \mathbf{v}_{n+1}$, $\mathbf{d}_{n+1}^{(1)} = \mathbf{d}_{n+1}$ and $\mathbf{f}_{n+1}^{(1)} = \mathbf{f}_{n+1}$ from (82), (83) and iterate over $i = 1, \cdots i_{max}$:

$$\mathbf{d}_{n+1}^{(i+1)} = \mathbf{d}_{n+1}^{(i)} + \beta_3 \Pi_3 \eta_3 \Delta t^2 \left(\dot{\mathbf{v}}_{n+1}^{(i)} - \dot{\mathbf{v}}_{n+1}^{*} \right) \tag{84}$$

$$\mathbf{v}_{n+1}^{(i+1)} = \mathbf{v}_{n+1}^{(i)} + \eta_3 \Delta t \left(\dot{\mathbf{v}}_{n+1}^{(i)} - \dot{\mathbf{v}}_{n+1}^{*} \right)$$

$$\mathbf{f}_{n+1}^{(i+1)} = \mathbf{f} \left(\mathbf{d}_{n+1}^{(i+1)}, \mathbf{v}_{n+1}^{(i+1)} \right)$$

$$\dot{\mathbf{v}}_{n+1}^{(i+1)} = \mathbf{M}^{-1} \left(\mathbf{p}_{n+1} - \mathbf{f}_{n+1}^{(i+1)} \right)$$

Velocity dependent internal forces are neglected here for simplicity. The multi- corrector steps (84) change the type of the overall scheme (79) - (83). Because of the iterative

improvement of the solution the scheme is no longer purely explicit. Velocity dependent internal forces are neglected for simplicity in the following considerations. The suggested iteration is a simple function iteration

$$\mathbf{z}^{i+1} = \mathbf{w}(\mathbf{z}^i) \tag{85}$$

$$\text{with } \mathbf{z}^i := \mathbf{d}^i_{n+1} ,$$

$$\text{and } \mathbf{w}(\mathbf{z}^i) := \mathbf{d}^{(i)}_{n+1} + \beta_3 \Pi_3 \eta_3 \Delta t^2 [\, \mathbf{M}^{-1} (\mathbf{p}_{n+1} - \mathbf{f}(\mathbf{d}^{(i)}_{n+1})) - \dot{\mathbf{v}}^{*}_{n+1}]$$

which converges if the constant Q defined by

$$|| \mathbf{w}(\mathbf{z}^{i+1}) - \mathbf{w}(\mathbf{z}^i) || \le Q || \mathbf{z}^{i+1} - \mathbf{z}^i || \tag{86}$$

remains less than one

$$Q \le 1 \tag{87}$$

The derivative of the function $\mathbf{w}(\mathbf{z})$ with respect to $\mathbf{z}$ gives a good estimate for the constant Q. If we approximate the nonlinear system (1) through the linearized system (73) in the time interval $t_n \le t \le t_{n+1}$ and decouple the system (73) with the actual eigenvalues Λ and eigenvectors Φ from (55) and (56) we can express the constant Q more precisely in terms of the actual eigenvalues. After the equations (85) and (86) are premultiplied by $\Phi^T \mathbf{M}$ and all state vectors are transformed with (58) we get the estimate for the constant Q

$$Q \le || \mathbf{I} - \Lambda \Delta t^2 \beta_3 \Pi_3 \eta_3 || \le 1 \tag{88}$$

Inserting (38) we get from (88) a convergence criteria for the iteration scheme (84) :

$$\Omega^2_{con} \le \frac{2}{\beta_3 \Pi_3 \eta_3} \tag{89}$$

with $\Omega_{con} = \omega_{max} \Delta t$ and ω_{max} denotes the highest eigenfrequency of the linearized system in the interval $t_n \le t \le t_{n+1}$. For all methods within the generalized algorithm the convergence criteria (89) and the stability limit (see table 2) lead to values which are close together $\Omega_{con} / \Omega_{crit} \approx 1$. To get a good convergence behavior of the iteration (84) the constant Q must be much smaller than one. Consequently time steps must be used which lie considerably below the criteria (89) that means also below the stability limit. Such time step sizes are only used if higher accuracy should be achieved or if the Courant- Friedrichs-Levy condition in wave propagation problems must be fulfilled. The experience with the iteration scheme (84) shows that a lot of iteration steps are necessary to improve the solution considerably. The internal forces must be evaluated once in each iteration step which

costs about a half of a full time step. Therefore the iteration can not be recommended in most cases.

5.3 Remarks to the Spectral Properties

Before giving numerical examples, we present some spectral curves to give a theoretical insight in the behavior of the methods. The spectral properties of the amplification matrix can provide some useful information , for example the curves of the spectral radius and of the third root over the frequency range. The period error and the numerical damping are artificially defined values from the spectral properties. They must be interpreted carefully because their definition depends on assumptions like complex conjugate roots and a small spurious root, see [14] for details. In a strict sense they are only valid for linear multistep methods. In the case of nonlinear methods the definitions for the relative period error and the numerical damping are only valid in terms of absolute values of these quantities.

Figure 1 shows the spectral radius of the new methods. For the methods with numerical damping the spectral radius starts with $\rho = 1$, then decreases to $\rho < 1$ but goes back to one before the stability limit. Thus, the numerical damping does not work in the high frequency range close to the stability limit. This behavior is common to all explicit methods with numerical damping. The numerical damping of the algorithm EX4D is very mild compared to the algorithm EX3D, see figure 1b. The higher order methods expose a very small period error compared to the central difference method, see figure 2. The period error of all methods increases substantially before the stability occurs.

6. NUMERICAL EXAMPLES

All examples are calculated with seven different methods, see Table 3. The central difference method and the methods of Katona and Zienkiewicz [10] need only one function evaluation per time step, the new higher order family developed in this paper needs two. Therefore the time step for the new methods is doubled to get a realistic comparison with respect to costs of the other methods.

6.1 Linear Undamped Single Degree of Freedom System

To verify the theoretical results from the previous sections, accuracy is measured with respect to the calculation of a simple undamped oscillator under free vibrations

$$\ddot{u} + \omega^2 u = 0 \tag{90a}$$

subject to initial conditions

$$u(0) = 1.0, \quad \dot{u}(0) = 0.0 \tag{90b}$$

In numerical calculations the oscillator has a period $T = 1.0$ and an frequency $\omega = 2\pi$. In figure 3 the error in the displacements at the end of the fourth period is plotted in a double logarithmic scale. The slope of the error curve provides directly the practical rate of convergence. The second and third order methods all have the same asymptotic rate which is approximately 3, see figure 3a. The estimate of the global truncation error (41) turns out to be conservative for a symmetric method like central difference, which is well known. The fourth order methods show a convergence rate of about 5, see figure 3b. To make comparisons easier the fourth order methods are always plotted together with the second order version NF2, see figure 3b.

The difference between the second and the third order methods is evident after several oscillations, see the displacements after 20 periods in figure 4a and after 40 periods in figure 5a. All third order methods considered here possess numerical damping which may be desirable in some cases. The lower period error makes the third order methods more attractive than second order methods. Among the fourth order methods the scheme of Katona and Zienkiewicz is the most accurate after 20 periods, see figure 4b. After 40 and 80 periods the fourth order versions NF4 and EX4 developed in this paper are superior to the other methods, see figure 5b and figure 6. All fourth order methods have no numerical dissipation. The big gain in accuracy of the fourth order methods compared to second order is exposed in figure 6. The fourth order methods are able to follow the period very accurately over a long period of time.

In multidegree of freedom systems the time step is chosen very close to the stability limit with respect to the highest eigenfrequency of the system. The numerical method need not amplify these higher frequencies. Especially in the beginning of the calculation a reliable method must be free of oscillations. Therefore the behavior of all methods is investigated for time step sizes close to the stability limit. In figure 7 the absolute value of the displacements in the first 16 and 32 time steps is plotted. All methods amplify the

displacements by no more than a factor of 2 which is sufficiently small for most multidegree of freedom systems. The third order version of Katona E3D shows the most reliable numerical damping, see figure 7a.

6.2 Hardening Spring

A geometric nonlinear hardening spring due to a two bar snap through problem is tested, see figure 8. This example has been used by several authors, and it is very convenient to show the accuracy of step- by- step integration methods in a nonlinear case, see for example [11] and [24]. The undamped free vibration of the hardening spring with an initial displacement is described by the following equations:

$$m\,\ddot{u}(t) + f(u(t)) = 0 \qquad (91)$$

$$u(0) = u_0, \qquad \dot{u}(0) = 0$$

$$f(u) = 2\left\{ S\,\frac{u}{(l^2 + u^2)^{1/2}} + EF\left[\frac{u}{l} - \frac{u}{(l^2 + u^2)^{1/2}}\right]\right\} \qquad (92)$$

A calculation with very short time steps converges to the quasi exact period of the system $T = 0.264\ 791\ 269$ s. In every period the mass must return to the initial position $u_0 = 0.2$ m. This criteria is used to define the error in the numerical solution.

The convergence rates in the displacements at the end of the fourth period for this special example are shown in figure 9. All higher order methods are still more accurate than the second order methods but loose the higher convergence rate of their linear counterparts. This behavior will occur in all the following nonlinear examples. To maintain the accuracy of the higher order methods the evaluation of a tangent stiffness would be necessary. This should be avoided here to take full advantage of the explicit nature of the schemes. Nevertheless it is worthwhile to apply the higher order schemes because they show a lower error level and a better long time behavior compared to second order methods.

The long time behavior is exposed by plotting the error in the displacements at the end of each period over 40 cycles, see figure 10. The second order version NF2 and Katona's fourth order version E4 show the most accurate behavior in this example.

For time steps close to the stability limit some methods are no longer reliable, see the central difference method and Katona's fourth order version E4 in figure 11. It is well known that these instabilities may occur because the stability limit can be exceeded in stiff regions of the hardening spring.

6.3 Free Vibration of a Softening Spring

The undamped free vibration of a softening spring with initial conditions is calculated from

$$\ddot{u}(t) + 100 \tanh(u(t)) = 0 \qquad (93)$$

$$u(0) = 0 \qquad \dot{u}(0) = 25.$$

The same example was used by Park [31]. The numerical solution converges to the quasi exact period T = 1.116 830 s with a peak amplitude of $u_{max} = 3.8177$.

The rates of convergence for the higher order methods are again lower than in the linear case, see the convergence in the displacements at the end of four periods in figure 12. In the long time behavior higher order methods offer some advantages, see the error in the displacements after each period over 40 cycles in figure 13. The third and fourth order versions of Katona are the most accurate methods in this example. All methods show an unstable behavior for time steps close to the stability limit computed using the initial (maximum) stiffness, see figure 14. In spite of softening in the system the long term stability is not maintained, since the large displacement increments in the softening branch destroy the estimate based upon the linearized system, see equation (74). Furthermore the change from constant to varying internal forces causes an accumulation of energy in the numerical solution.

6.4 Forced Vibration of a Softening Spring

The undamped forced vibration of a softening spring with zero initial conditions is calculated from

$$\ddot{u}(t) + \tanh(u(t)) = p_0 \qquad (94)$$

$$u(t) = 0 \qquad \dot{u}(t) = 0$$

This example was also used by Kujawski and Gallagher [32]. For a constant external load $p_0 = 0.75$ the numerical solution converges to the quasi exact period T = 11.5866 s and a peak amplitude of $u_{max} = 2.75647$.

The convergence rate in the displacements shows the same behavior as in the foregoing examples except that the third and fourth order versions of the generalized algorithm developed in this paper expose very low error levels in the region of $\Delta t/T \approx 0.01$, see figure 15. Looking at the long time behavior in figure 16 the third and fourth order versions of the generalized algorithm are the most accurate methods in this example. For larger time steps $\Delta t/T=1/10$ nd1/20 the higher order methods do not provide better solutions than the lower order methods, see figure 17. Again all methods expose an unstable behavior for time steps close to the stability limit, especially the central difference method, Kujawski's NF2 version and Katona's E4 version, see figure 18. This appears to be caused by large displacement increments as in the foregoing example.

6.5 Bilinear Softening Spring

The free vibration of a bilinear softening spring shown in figure 19 with an initial velocity is calculated:

$$\ddot{u}(t) + f(u(t)) = 0 \tag{95}$$

$$u(0) = u_0 = 0 \qquad \dot{u}(0) = \dot{u}_0 = 25$$

$$f(u) = \begin{cases} 100\,u & |u| \le 2 \\ 200 & |u| > 2 \end{cases} \tag{96}$$

The numerical solution converges to the quasi exact period $T = 0.670\,918$ s and the peak amplitude $u_{max} = 2.5625$. The difficulties of this example arise from the nondifferentiable internal forces f(u). The problem was investigated by several authors mainly with implicit schemes, see e.g. [1], [11] and [33]. Higher order methods which use the derivative of the internal forces must certainly loose their accuracy at the nondifferentiable point. From the mathematical theory the methods must remain stable since the internal forces are still Lipschitz continuous.

The convergence rates in the displacements expose the fact that no gain in accuracy can be expected from higher order methods in this special example, see figure 20. In the long time behavior the error in the displacements shows a cyclic curve for the higher order methods, see figure 21. In the average the higher order methods are as accurate as the lower order methods. The same behavior can be observed for larger step sizes, see figure 22. Obviously the introduction of numerical damping does not improve the solution in

this case, see figure 22a. For time step sizes close to the stability limit the versions of the generalized algorithm developed in this paper show some instabilities, see figure 23. The versions of Katona behave quite reliably in this example.

7. CONCLUSIONS

The developed concept of a generalized explicit one step algorithm based on the idea of higher derivative methods leads to some useful third and fourth order schemes in non-linear structural dynamics. The algorithm may be implemented in a predictor- corrector form with two evaluations of the internal forces per time step. The stability limit is nearly twice as large as for the second order accurate central difference method which needs one evaluation of the internal forces per time step. Thus, the computational effort increases by about 10% compared to the lower order methods. A few more vector operations must be carried out and the stability limit decreases slightly. No matrix vector multiplications or further evaluations of the internal forces are necessary.

The small amount of increase in computational effort makes the algorithms attractive for finite element calculations. The gain in accuracy is large in linear and in mildly non-linear systems. For strong nonlinearities the convergence rates of the higher order methods can decrease to second order. Without additional evaluations of the internal force vector we can not achieve better convergence rates. It is still worthwhile to use the higher order methods, even in strongly nonlinear systems, since the error level and the long time behavior is in general better than those for the lower order methods.

To insure the stability of the algorithm in nonlinear systems the internal forces, the displacement increments and the energy must be controlled. A pure time step control with the critical time step derived from the linearized system does not prevent instabilities caused by sudden stiffening or complete softening.

All nonlinear methods based on the concept of higher derivatives or linear methods based on Taylor series expansion need higher derivatives of the accelerations to achieve higher accuracy. In some practical examples nondifferentiable internal forces may occur. As long as these forces satisfy Lipschitz continuity the higher order methods are stable and maintain at least second order accuracy. This has been proved theoretically and has been demonstrated in a numerical example.

The methods developed in this paper are viable alternatives to the methods of Katona and Zienkiewicz [10]. In linear systems both methods behave similarly. In nonlinear systems the methods from [10] show better results in free vibration whereas the methods from the present paper give better results in forced vibration. In nonlinear systems the quality of the results varies from one example to another and one may always find an example were a particular development looks better than alternative methods.

Acknowledgements

The first author appreciates many enjoyable discussions with Prof. L. Vu- Quoc and Dr. M. Olsson during the conduct of this research work and special thanks are extended to Prof. L. Vu- Quoc for some valuable hints to the convergence proof. The support from the German foundation "Deutscher Akademischer Austauschdienst" within the NATO post-doctoral program is gratefully acknowledged.

REFERENCES

[1] Belytschko, T.; Hughes, T.J.R.; *Computational Methods for Transient Analysis* ,Vol.1 in *Computational Methods in Mechanics* , North Holland Elsevier Publishers, Amsterdam (1983)

[2] W.L. Wood, *Some Transient and Coupled Problems - A State- of- the- Art- Review* , Chapter 8 in *Numerical Methods in Transient and Coupled Problems* , Ed. R.W. Lewis, E. Hinton, P. Bettess and B.A. Schrefler, John Wiley & Sons Ltd. (1987)

[3] E.Hairer, S.P.Nørsett, G.Wanner, *Numerical Solution of Ordinary Differential Equations* , Springer Series in Computational Mathematics, Springer Verlag Berlin, New York (1987)

[4] G.Gupta, R.Sacks-Davis, P.E.Tischer, *A Review of Recent Developments in Solving ODEs* ,Computing Surveys, Vol.17 No.1 (1985)

[5] P.Deuflhard, *Recent Progress in Extrapolation Methods for Ordinary Differential Equations* , SIAM Review, Vol.27 No.4 (1985)

[6] K.Dekker, J.G. Vekker, *Stability of Runge- Kutta Methods for Stiff Nonlinear Differential Equations* ,CWI Monographs, North Holland Elsevier Publishers (1984)

[7] C.C. Fu, *A Method for the Numerical Integration of the Equation of Motions Arising from a Finite Element Analysis* ,ASME J. Appl. Mech., Sept. 70, 599- 605 (1970)

[8] J.Kujawski, R.H.Gallagher, *A Family of Higher- Order Explicit Algorithms for the Transient Dynamic Analysis* Trans. of the Society for Computer Simulation, Vol.1 No.2, 155- 166 (1984)

[9] J.Kujawski, C.Miedzialowski, L.Krasuski, R.H.Gallagher, *Analysis of Higher Order Explicit Time Integration Operators for Nonlinear Problems* ,Proceedings of the NUMETA '87, Swansea (1987)

[10] M.G.Katona, O.C.Zienkiewicz, *A Unified Set of Single Step Algorithms, Part 3 : The Beta- m Method, a Generalization of the Newmark Scheme* , Int. J. Num. Meth. Eng., Vol.21, 1345- 1359 (1985)

[11] M. Gellert, *A Direct Integration Method for the Analysis of a Certain Class of Nonlinear Dynamic Problems* , Ing.Archiv 48, 403- 415 (1979)

[12] O.C. Zienkiewicz, W.L. Wood, N.W. Hine, R.L. Taylor, *A Unified Set of Single Step Algorithms, Part 1: General Formulation and Applications* , Int. J. Num. Meth. Eng., Vol.20, 1529- 1552 (1984)

[13] H.M. Hilber, T.J.R. Hughes, R.L. Taylor, *Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics* , Earthquake Eng.& Struct.Dyn., Vol.5, 283- 292 (1977)

[14] C. Hoff, P.J. Pahl, *Development of an Implicit Method with Numerical Dissipation from a Generalized Single Step Algorithm for Structural Dynamics* , Comp. Meth. Appl. Mech. Eng. 67, 367- 385 (1988)

[15] C. Hoff, P.J. Pahl, *Practical Performance of the Θ_1- Method and Comparison with Other Dissipative Algorithms in Structural Dynamics* , Comp. Meth. Appl. Mech. Eng. 67, 87- 110 (1988)

[16] G. Corliss, Y.F. Chang, *Solving Ordinary Differential Equation Using Taylor Series* , ACM Trans. Math. Softw. 8, Vol.2 June, 114- 144 (1982)

[17] S.P. Nørsett, *One Step Methods of Hermite Type for the Numerical Integration of Stiff Systems* , BIT Vol.14, 63- 77 (1974)

[18] W.H. Enright, *Optimal Second Derivative Methods for Stiff Systems,* in *Stiff Differential Systems,* R. Willoughby, Ed.Plenum, New York, 95- 109 (1974)

[19] W.H. Enright, *On the Efficient Time Integration of Systems of Second Order Equations Arising in Structural Dynamics,* Int. J. Num. Meth. Eng., Vol.16, 13- 18 (1980)

[20] R. Sacks- Davis, L.F. Shampine, *A Type- Insensitive ODE Code Based on Second Derivative Formulas,* Techn. Report, Dept. Comp. Science, Monash University Australia (1981)

[21] C.A. Addison, I. Gladwell, *Second Derivative Methods Applied to Implicit First- and Second- Order Systems,* Int. J. Num. Meth. Eng., Vol.20, 1211- 1231 (1984)

[22] R.M. Thomas, I. Gladwell, *The Methods of Gellert and of Brusa and Nigro are Padé Approximant Methods*, Int. J. Num. Meth. Eng., Vol.20, 1307- 1322 (1984)

[23] P. Lancaster, *Lambda Matrices and Vibrating Systems*, Pergamon, Oxford (1966)

[24] J.H. Argyris, J. St. Doltsinis, W.C. Knudson, L.E. Vaz, K.J. Willam, *Numerical Solution of Transient Nonlinear Problems*, Comp. Meth. Appl. Mech. Eng., Vol. 17/18, 341- 409 (1979)

[25] D.M. Trujillo, *The Direct Numerical Integration of Linear Matrix Differential Equations Using Padé Approximations*, Int. J. Num. Meth. Eng., Vol.9, 259- 270 (1975)

[26] L. Brusa, L. Nigro, *A One Step Method for the Direct Integration of Structural Dynamic Equations*, Int. J. Num. Meth. Eng., Vol.15, 685- 699 (1980)

[27] S.M. Serbin, *On a Fourth Order Unconditionally Stable Scheme for Damped Second Order Systems*, Comp. Meth. Appl. Mech. Eng., Vol.23, 333- 340 (1980)

[28] O.C.Zienkiewicz, W.L.Wood, R.L.Taylor, *An Alternative Single- Step Algorithm for Dynamic Problems* ,Earthquake Eng.& Struct. Dyn. Vol.8, 31- 40 (1980)

[29] W.L.Wood, *Further Look at Newmark, Houbold etc. Time Stepping Formulae* , Int. J. Num. Meth. Eng., Vol.20, 1009- 1017 (1984)

[30] S.N.Penry, W.L.Wood, *Comparison of Some Single- Step Methods for the Numerical Solution of the Structural Dynamic Equation* ,Int. J. Num. Meth. Eng., Vol.21, 1941- 1955 (1985)

[31] K.C. Park, *An Improved Stiffly Stable Method for Direct Integration of Nonlinear Structural Dynamic Equations*, ASME J. Appl. Mech. June 75, 464- 470 (1975)

[32] J. Kujawski, R.H. Gallagher, *An Efficient Higher Order Algorithm for Nonlinear Transient Dynamic Analysis*, Proceedings of the NUMETA' 85, Swansea (1985)

[33] T.J.R. Hughes, *Stability, Convergence and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics*, Comp.& Struct., Vol.6, 313- 324 (1976)

[34] J.D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons (1977)

[35] I. Gladwell, R.M. Thomas, *A Qualitative Analysis of Two and Three Step Methods for Stable Second Order Systems*, Numerical Analysis Report No. 83, Dept.of Mathematics, University of Manchester (1984)

[36] C.W. Gear, *The Stability of Numerical Methods of Second Order O.D.E.'s*, SIAM J.N.A. 15, 188- 197 (1978)

List of Tables and Figures

Figure 12 : Convergence in the displacements after 4 periods
(free vibration of an undamped softening spring tanh u)

Figure 13 : Absolute error in the displacements at the end of each period over 40 cycles with time steps $\frac{\Delta t}{T} = \frac{1}{100}, \frac{1}{200}$ (free vibration of an undamped softening spring)

Figure 14 : Initial stability behavior in the displacements during the first 32 time steps $\frac{\Delta t}{T} = \frac{1}{4}$ (free vibration of an undamped softening spring)

Figure 15 : Convergence in the displacements after 4 periods
(forced vibration of an undamped softening spring tanh u)

Figure 16 : Absolute error in the displacements at the end of each period over 40 cycles with time steps $\frac{\Delta t}{T} = \frac{1}{100}, \frac{1}{200}$ (forced vibration of an undamped softening spring)

Figure 17 : Displacements after 8 periods with time steps $\frac{\Delta t}{T} = \frac{1}{10}, \frac{1}{20}$
(forced vibration of an undamped softening spring)

Figure 18 : Initial stability behavior in the displacements during the first 40 time steps $\frac{\Delta t}{T} = \frac{1}{4}$ (forced vibration of an undamped softening spring)

Figure 19 : Bilinear softening spring

Figure 20 : Convergence in the displacements after 4 periods
(undamped bilinear softening spring)

Figure 21 : Absolute error in the displacements at the end of each period over 40 cycles with time steps $\frac{\Delta t}{T} = \frac{1}{100}, \frac{1}{200}$ (undamped bilinear softening spring)

Figure 22 : Displacements after 20 periods with time steps $\frac{\Delta t}{T} = \frac{1}{16}, \frac{1}{32}$
(undamped bilinear softening spring)

Figure 23 : Initial stability behavior in the displacements during the first 48 time steps $\frac{\Delta t}{T} = \frac{1}{4}$ (undamped bilinear softening spring)

method	$\hat{\eta}_2 =$	$\hat{\eta}_3 =$	$\hat{\beta}_3 =$	$\hat{\Psi}_2 =$	$\hat{\Pi}_3 =$
CD	0	0	0	1	1
NF	$\hat{\eta}_3$	$\frac{\beta_3}{2}\left[\left(1+\frac{4}{\beta_3}\right)^{1/2}-1\right]$	β_3	$\frac{1}{1-\hat{\eta}_3}$	$\frac{\beta_3}{\beta_3-\hat{\eta}_3^2}$
EX4	η_3	η_3	$\frac{2}{\eta_3}\left(\frac{1}{24}-\eta_3^2\right)$	$\frac{1+\eta_3}{1-\eta_3}$	$\frac{1}{24\left(\frac{1}{24}-\eta_3^2\right)}-\eta_3$
EX3D	$\frac{7}{30}$	$\frac{1}{4}$	$\frac{341}{630}$	$\frac{2}{3}$	$\frac{39}{40}$

Table 1 : Parameter adjusting for the generalized algorithm

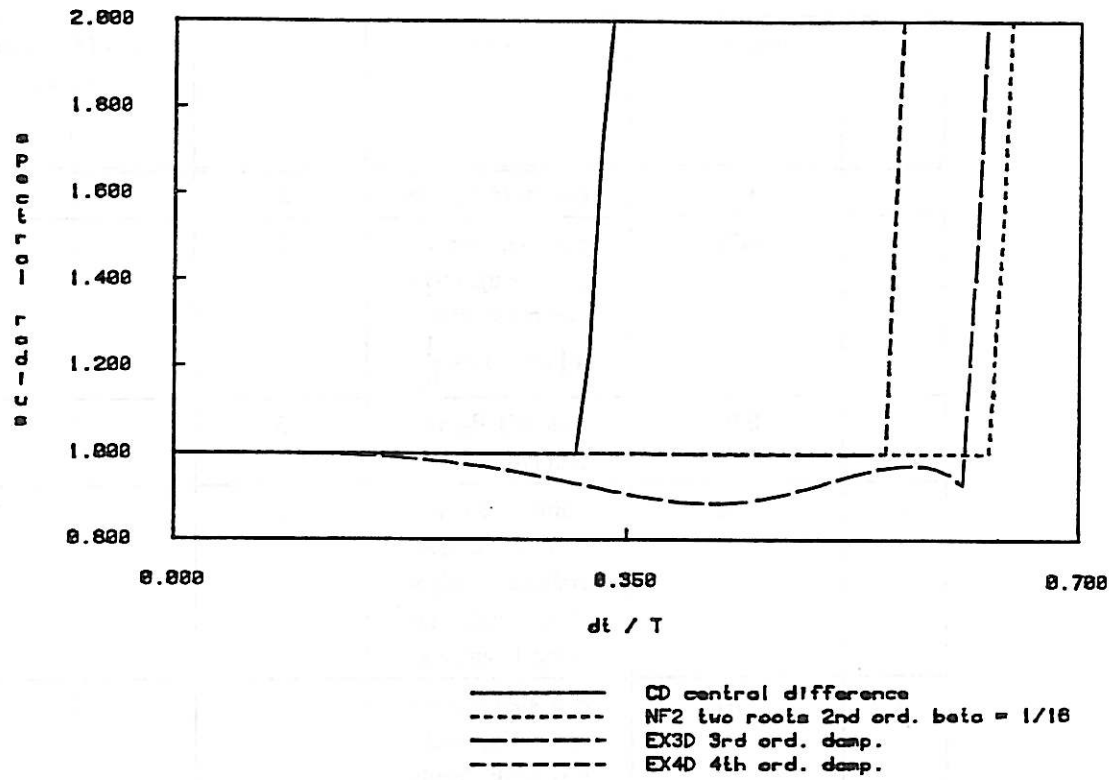
method	parameter value	accuracy p =	numerical damping	function evaluations n =	stability limit Ω_{crit}*	$\frac{\Omega_{crit}}{n}$	comments
CD		2	no	1	2	2	spectral identical to central difference
NF2	$\beta_3 = \frac{1}{16}$	2	no	2	4	2	spectral identical to the Newmark family
NF4	$\beta_3 = \frac{1}{12}$	4	no	2	$2\,(3)^{1/2} = 3.46$	1.73	of Kujawski, Gallagher et.al.[5]
EX4	$\eta_3 = \frac{1}{2}\left[\left(\frac{7}{6}\right)^{1/2} - 1\right]$	4	no	2	3.46	1.73	no numerical damping
EX4D	$\eta_3 = \frac{3}{8}\left[\left(\frac{7}{6}\right)^{1/2} - 1\right]$	4	mildly	2	3.46	1.73	mildly numerical damping
EX3D		3	yes	2	3.88	1.94	fixed numerical damping, maximal stability limit

* $\Omega_{crit} = \omega\,\Delta t_{crit} = 2\,\Pi\,\frac{\Delta t_{crit}}{T}$

Table 2 : Characteristics of the higher order methods covered by the generalized algorithm

no.	abbreviation	name	accuracy	no. of function evaluations n	stability limit $\frac{\Omega_{crit}}{n}$
1	CD	central difference	2	1	2
2	NF2	one step version of Kujawski's Newmark family [8] with $\beta_3=\frac{1}{16}$	2	2	2
3	E3D	Katona's β_3 version [10]	3	1	1.89
4	EX3D	Third order version of the generalized algorithm with numerical damping	3	2	1.94
5	NF4	one step version of Kujawski's Newmark family [8] with $\beta_3=\frac{1}{12}$	4	2	1.73
6	E4	Katona's β_4 version [10]	4	1	1.92
7	EX4	fourth order version of the generalized algorithm	4	2	1.73

Table 3: Methods applied to all examples



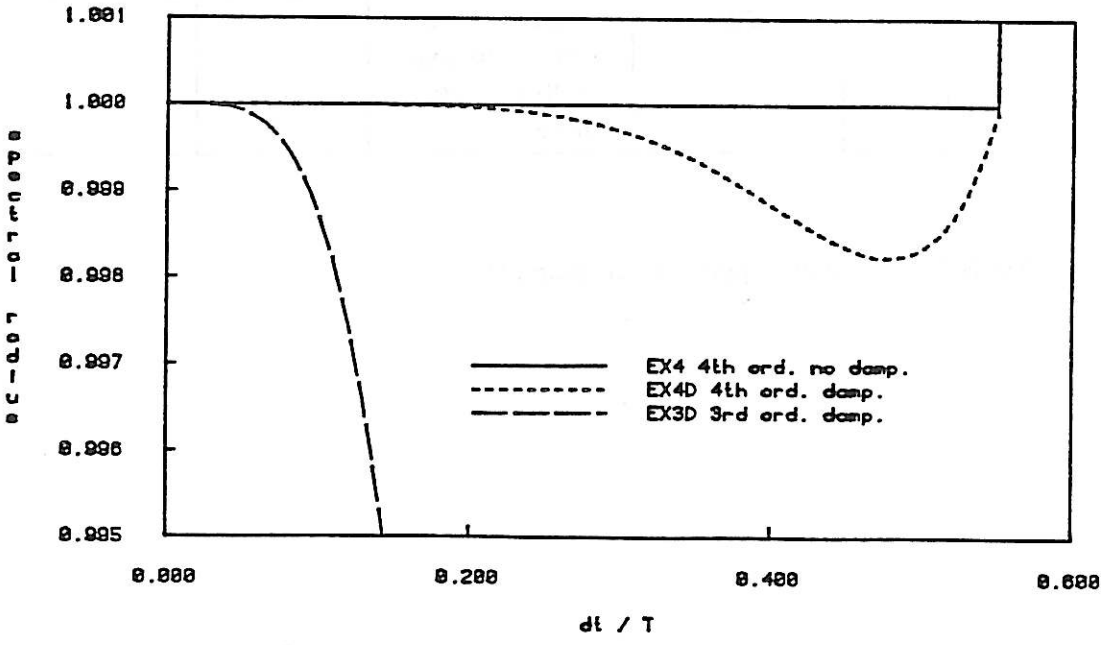


Figure 1 : Spectral radius of methods with second to fourth order accuracy derived from the generalized explicit algorithm

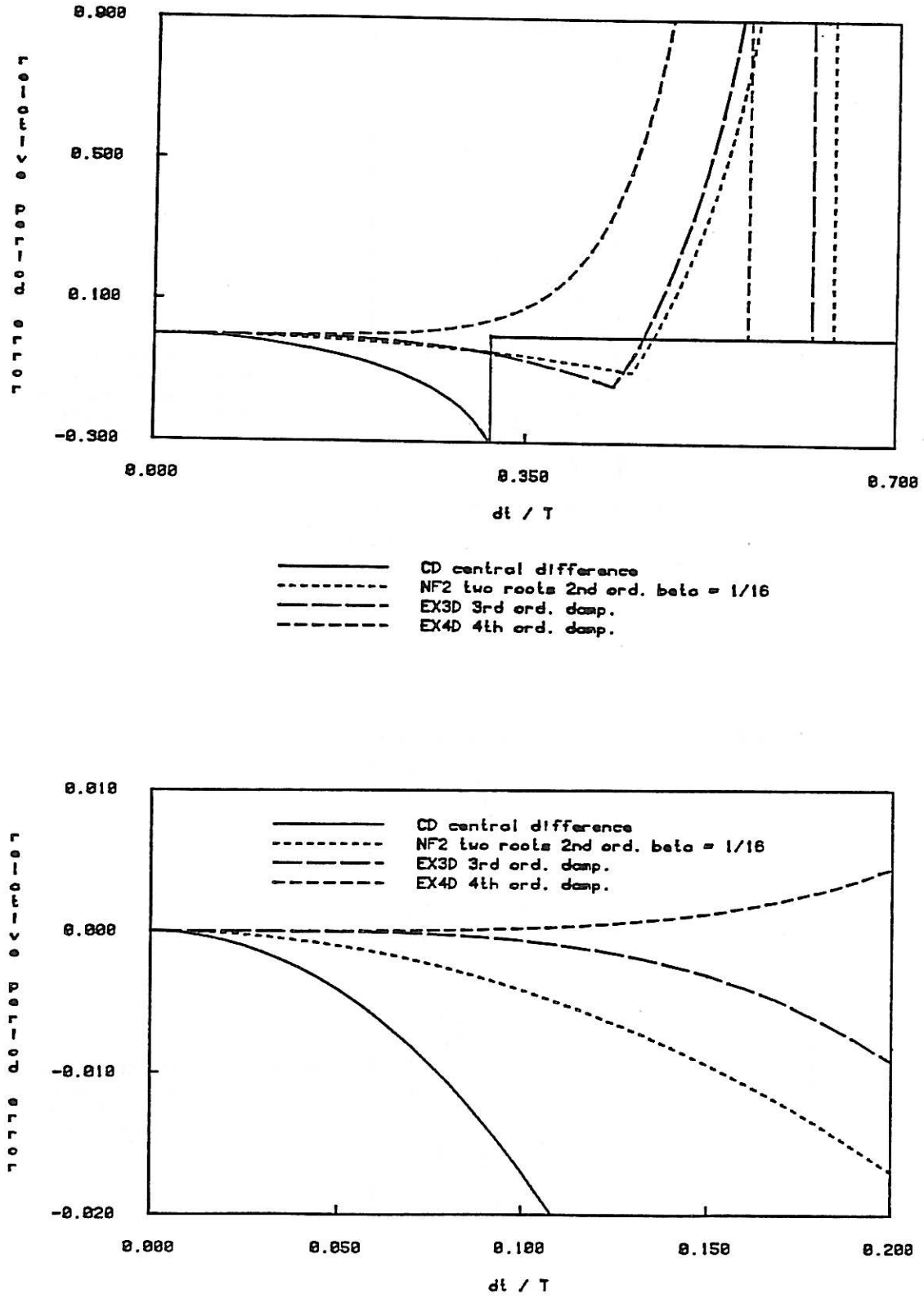


Figure 2 : Relative period error of methods with second to fourth order accuracy derived from the generalized explicit algorithm

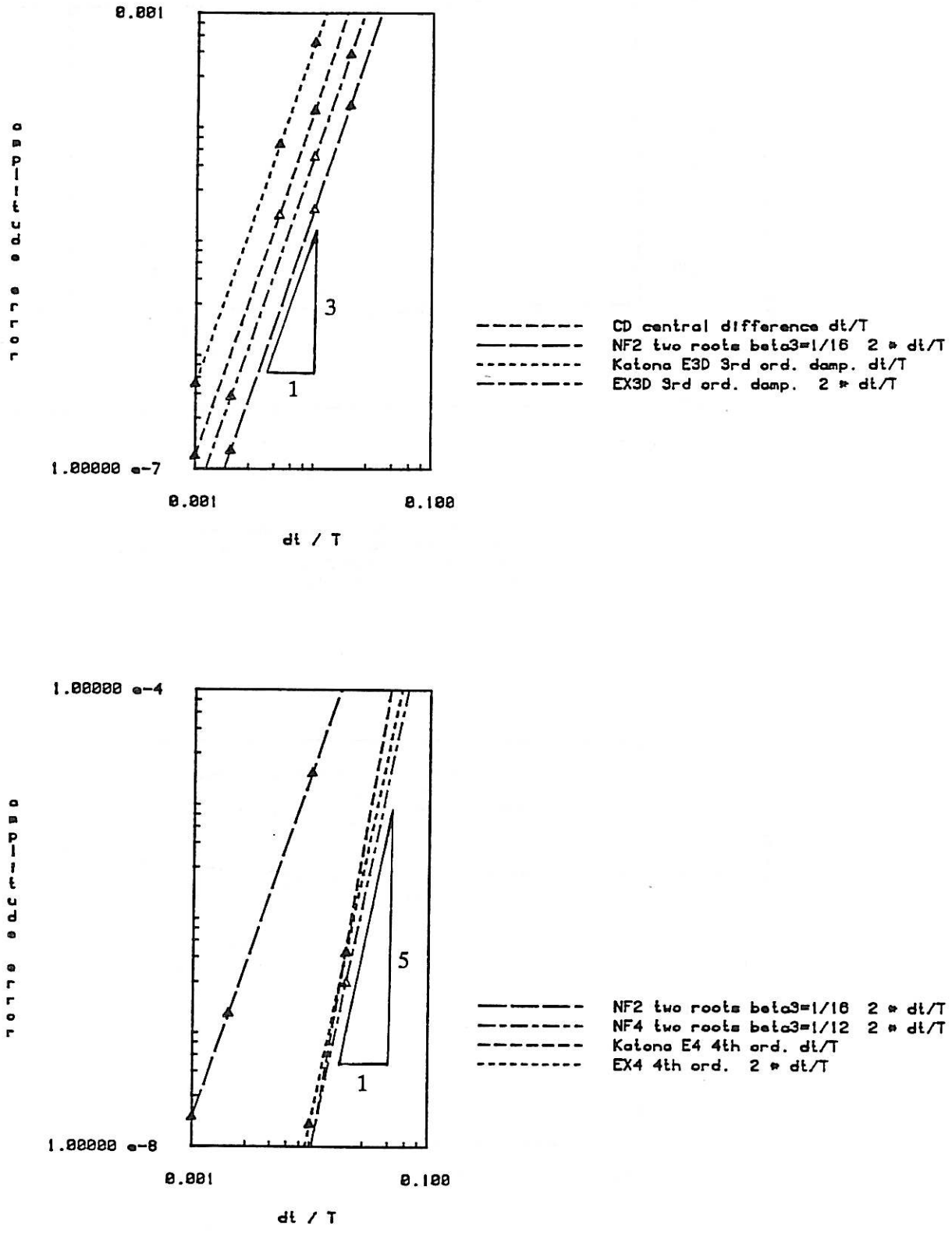
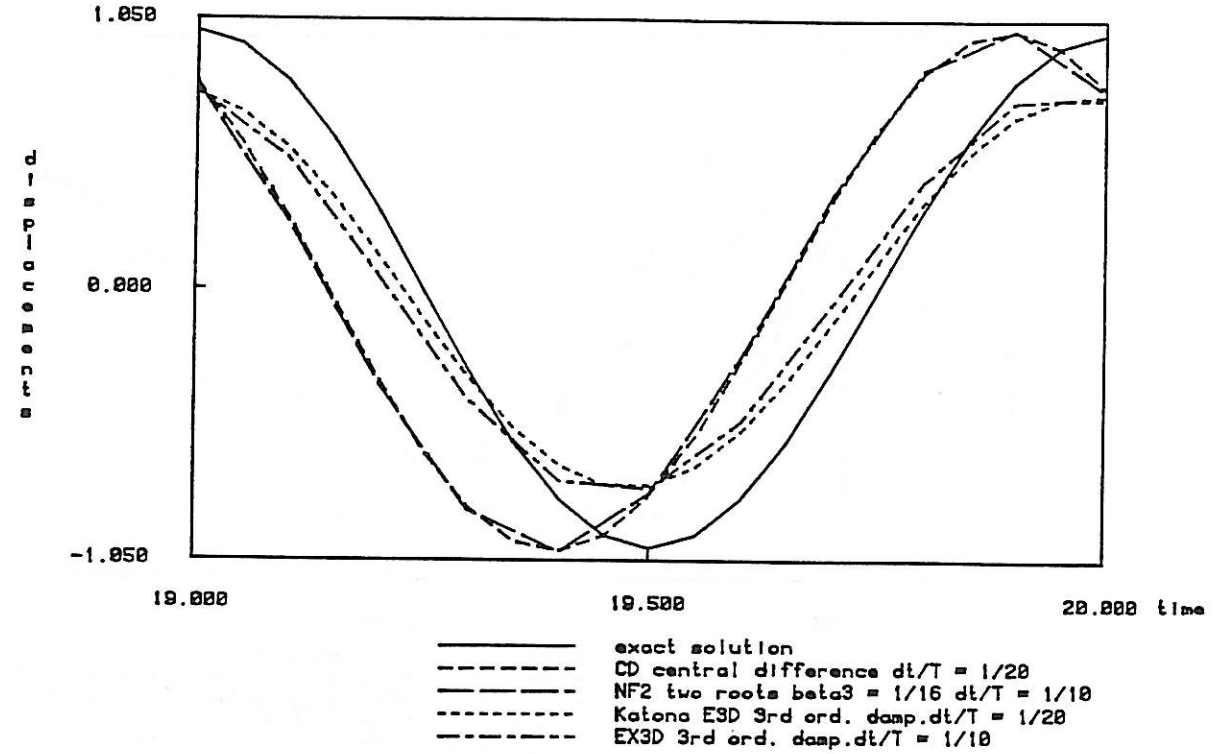


Figure 3 : Convergence in the displacements after 4 periods
(linear undamped single degree of freedom system)



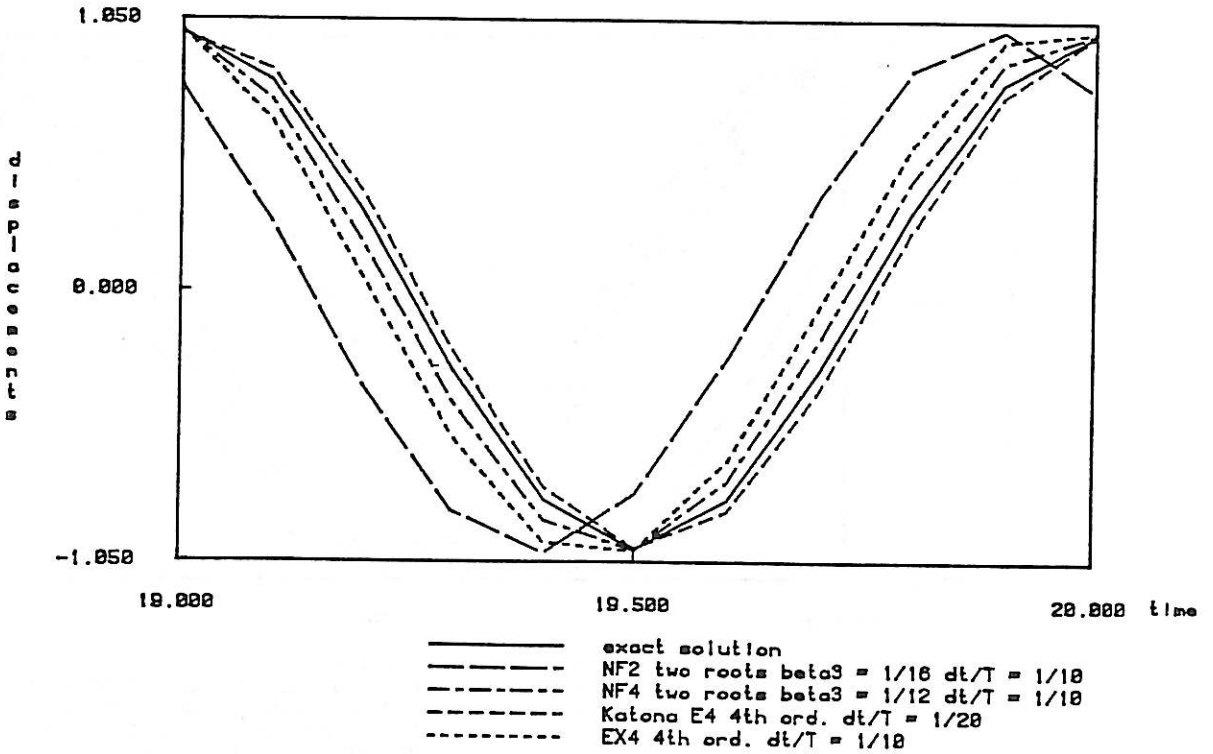


Figure 4 : Displacements after 20 periods for $\frac{\Delta t}{T} = \frac{1}{10}, \frac{1}{20}$

(linear undamped single degree of freedom system)

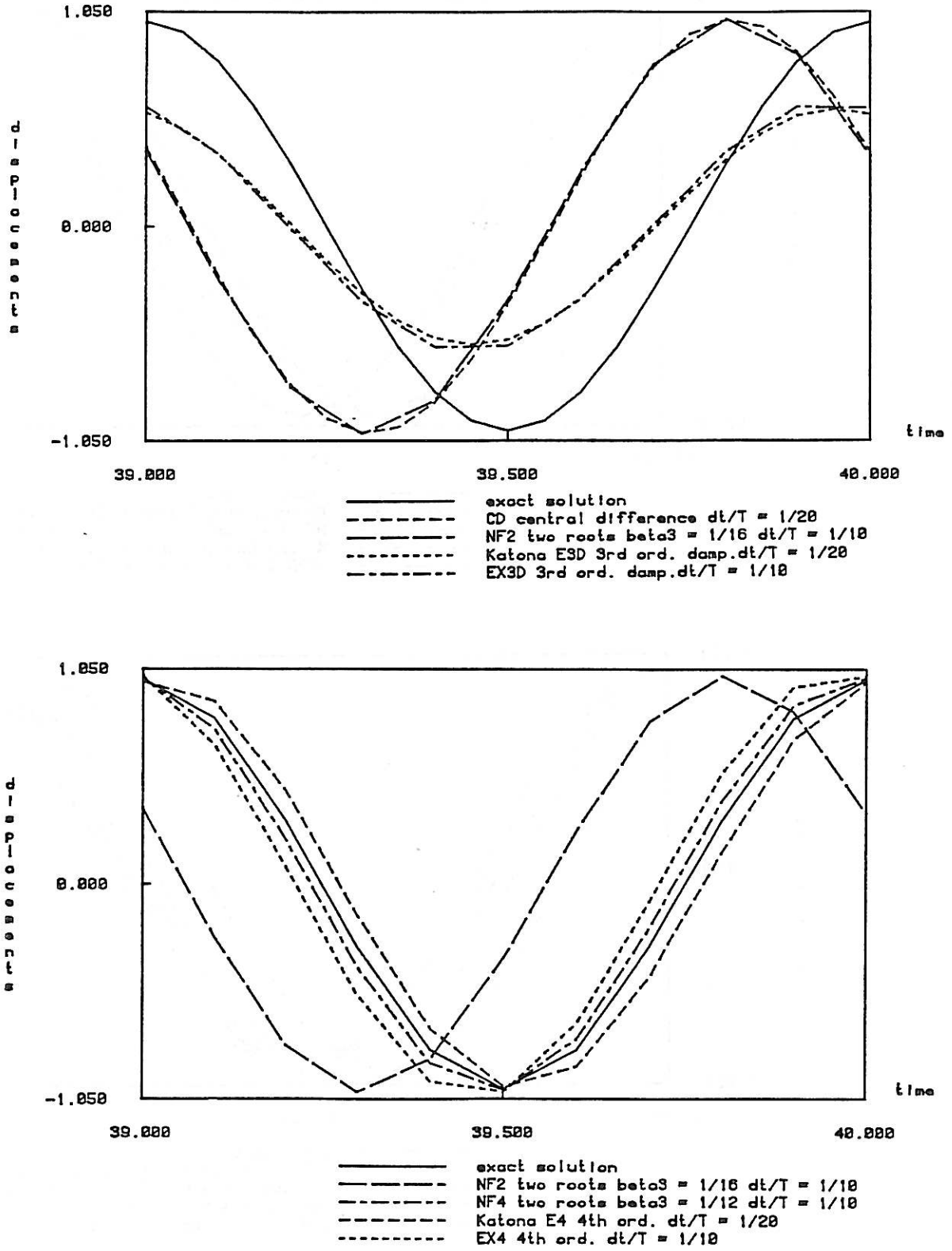


Figure 5 : Displacements after 40 periods for $\frac{\Delta t}{T} = \frac{1}{10}, \frac{1}{20}$

(linear undamped single degree of freedom system)

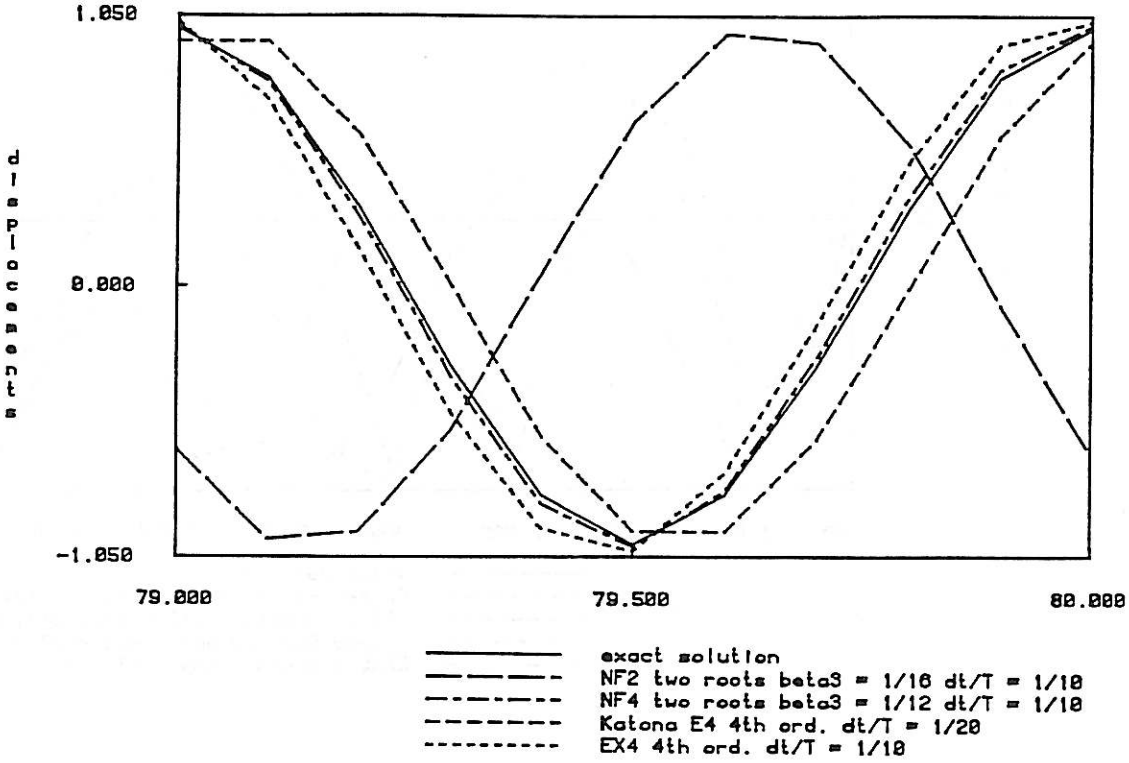


Figure 6 : Displacements after 80 periods for $\frac{\Delta t}{T} = \frac{1}{10}, \frac{1}{20}$

(linear undamped single degree of freedom system)

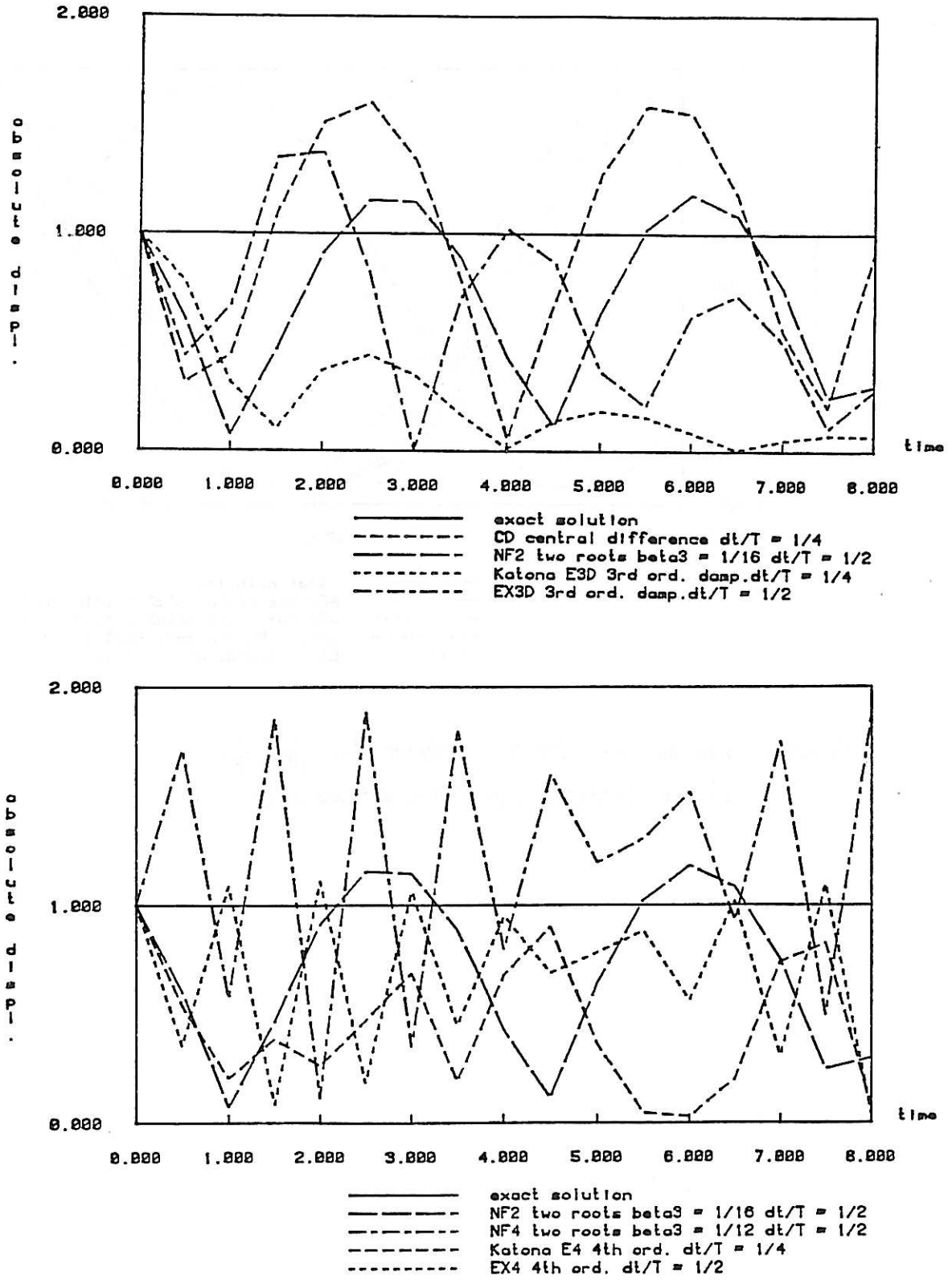


Figure 7 : Initial stability behavior in the absolute displacements during the first 16 resp. 32 time steps $\frac{\Delta t}{T} = \frac{1}{4}, \frac{1}{2}$ (linear undamped single degree of freedom system)

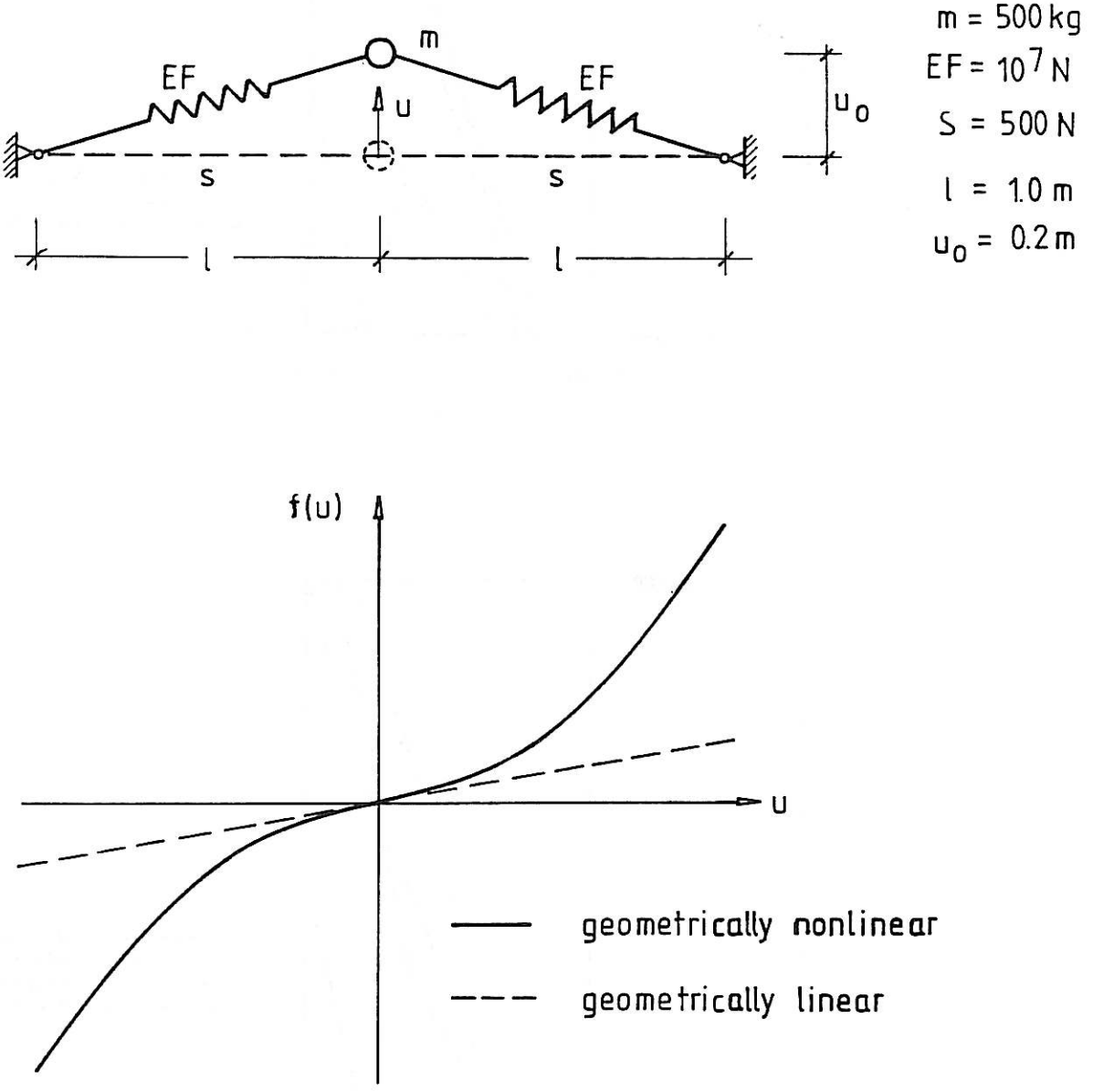


Figure 8 : Hardening spring due to a two bar snap through problem

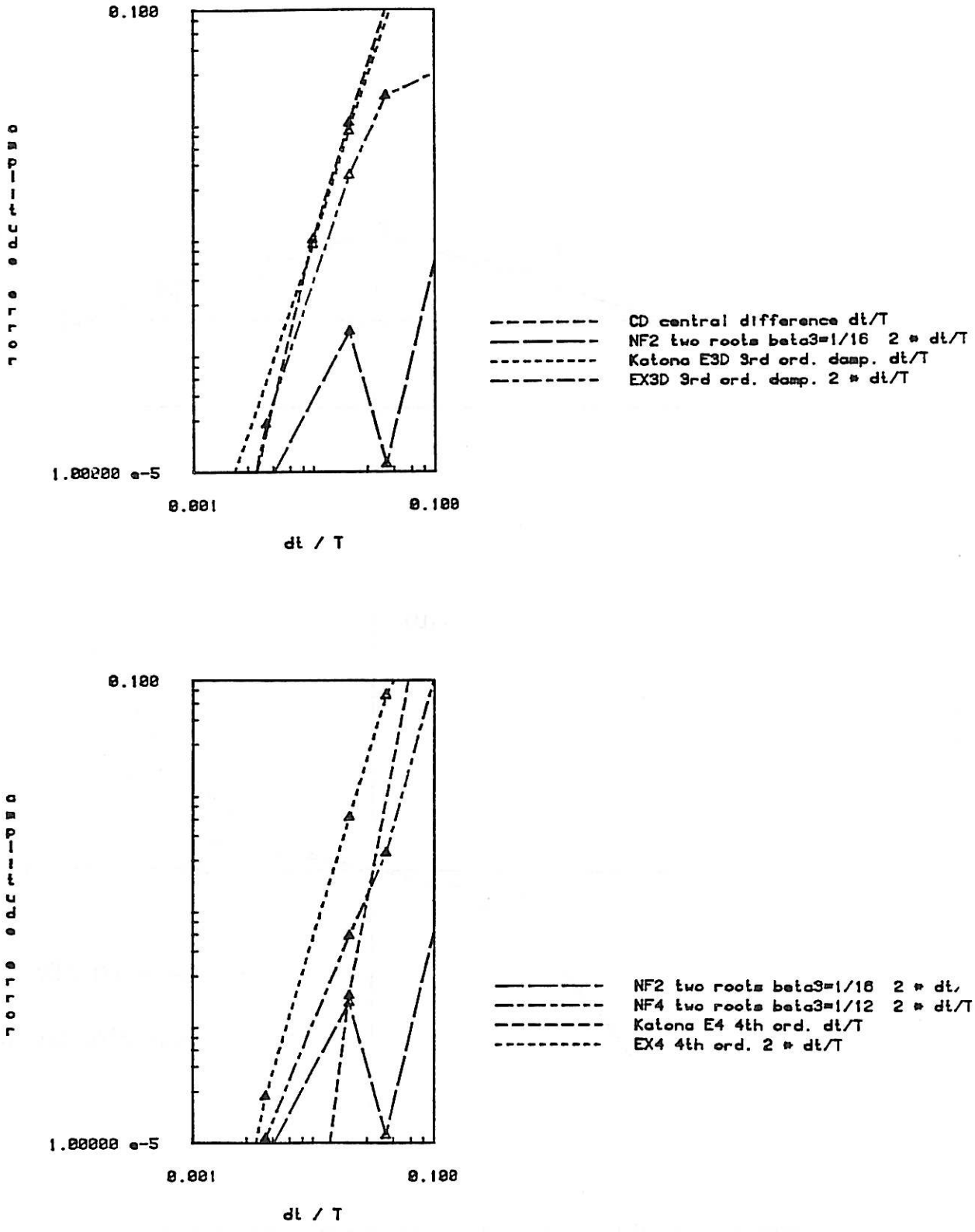
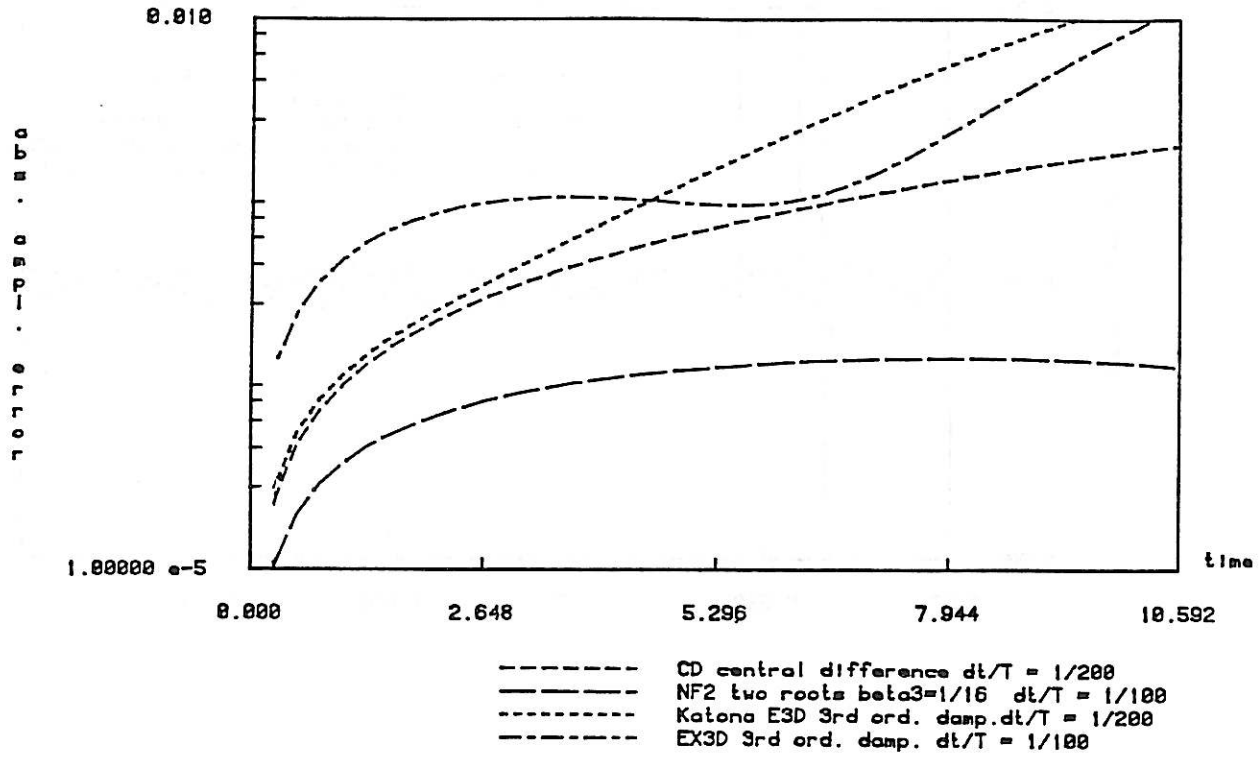


Figure 9 : Convergence in the displacements after 4 periods
(undamped hardening spring)



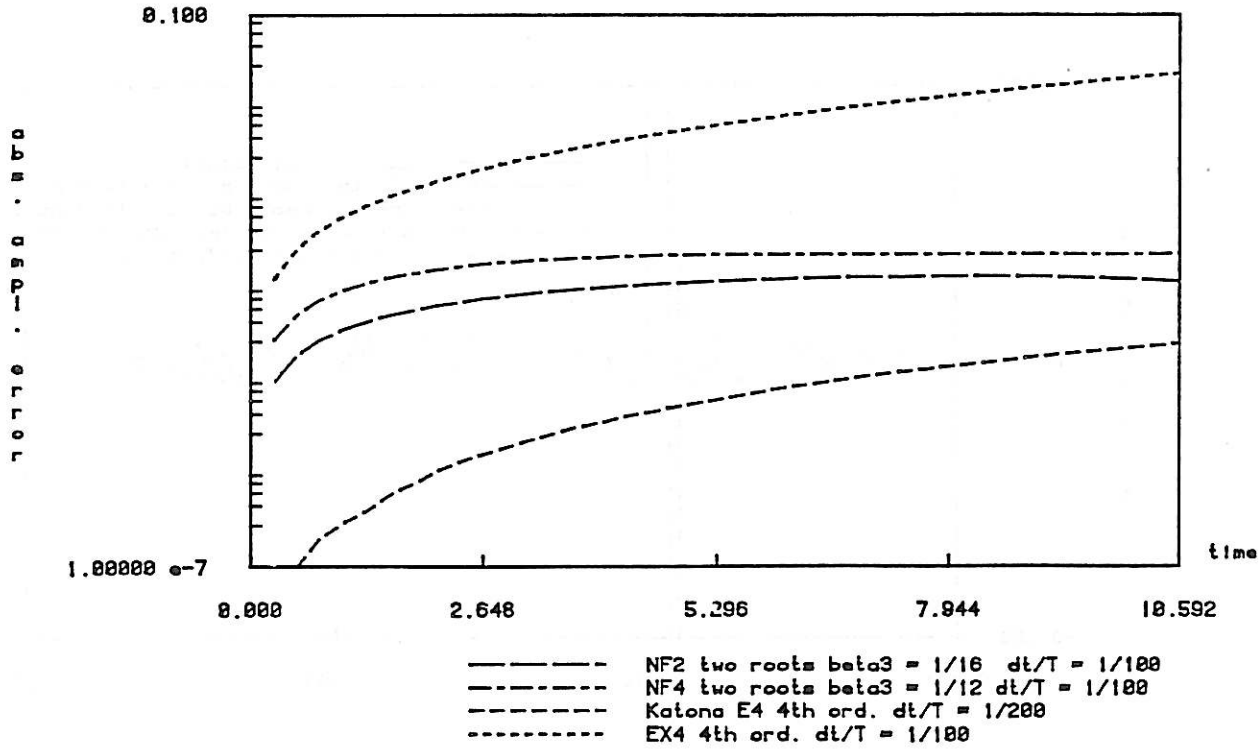
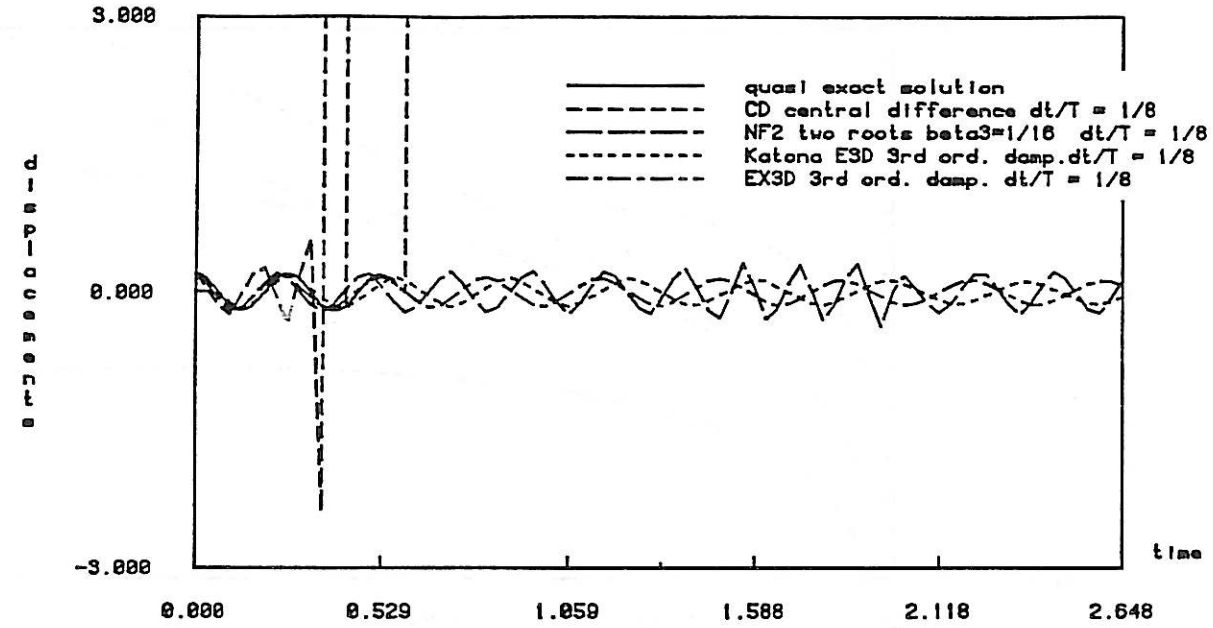


Figure 10 : Absolute error in the displacements at the end of each period over 40 cycles with time steps $\frac{\Delta t}{T} = \frac{1}{100}, \frac{1}{200}$ (undamped hardening spring)



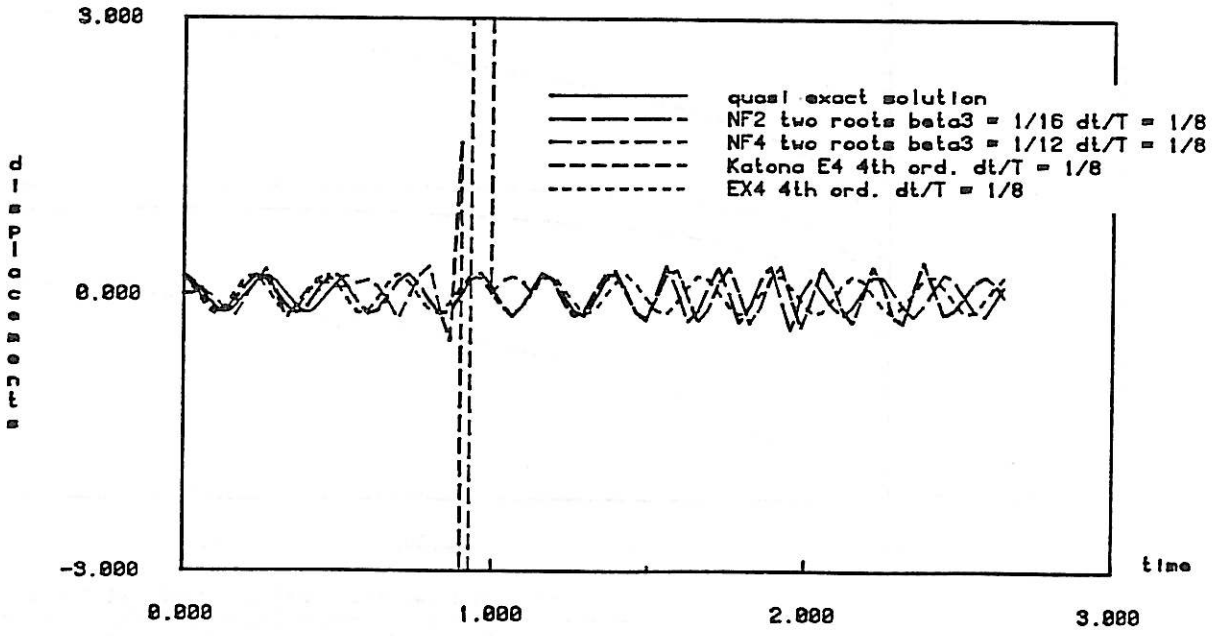


Figure 11 : Initial stability behavior in the displacements during the first 80 time steps $\frac{\Delta t}{T} = \frac{1}{8}$ (undamped hardening spring)

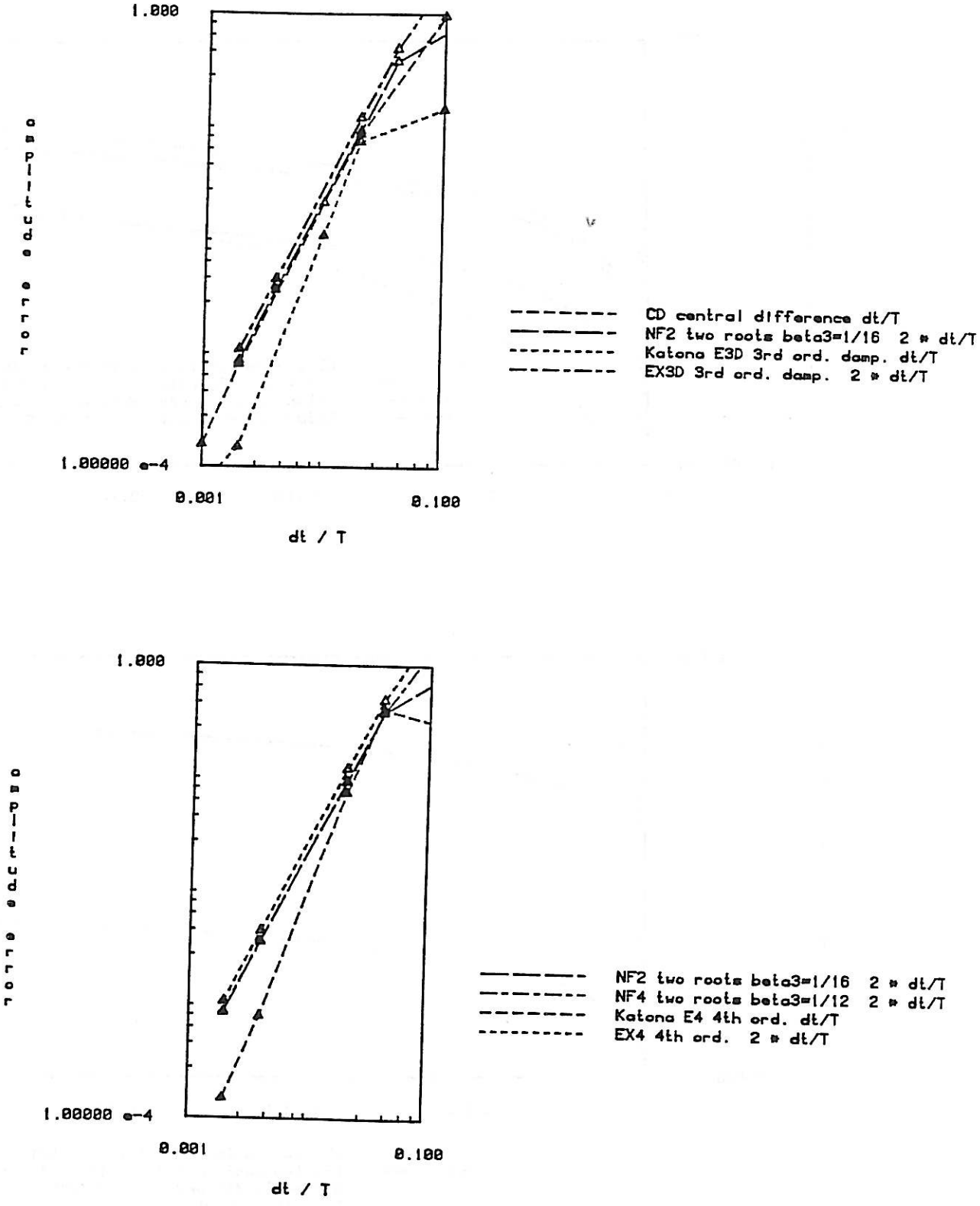


Figure 12 : Convergence in the displacements after 4 periods
(free vibration of an undamped softening spring tanh u)

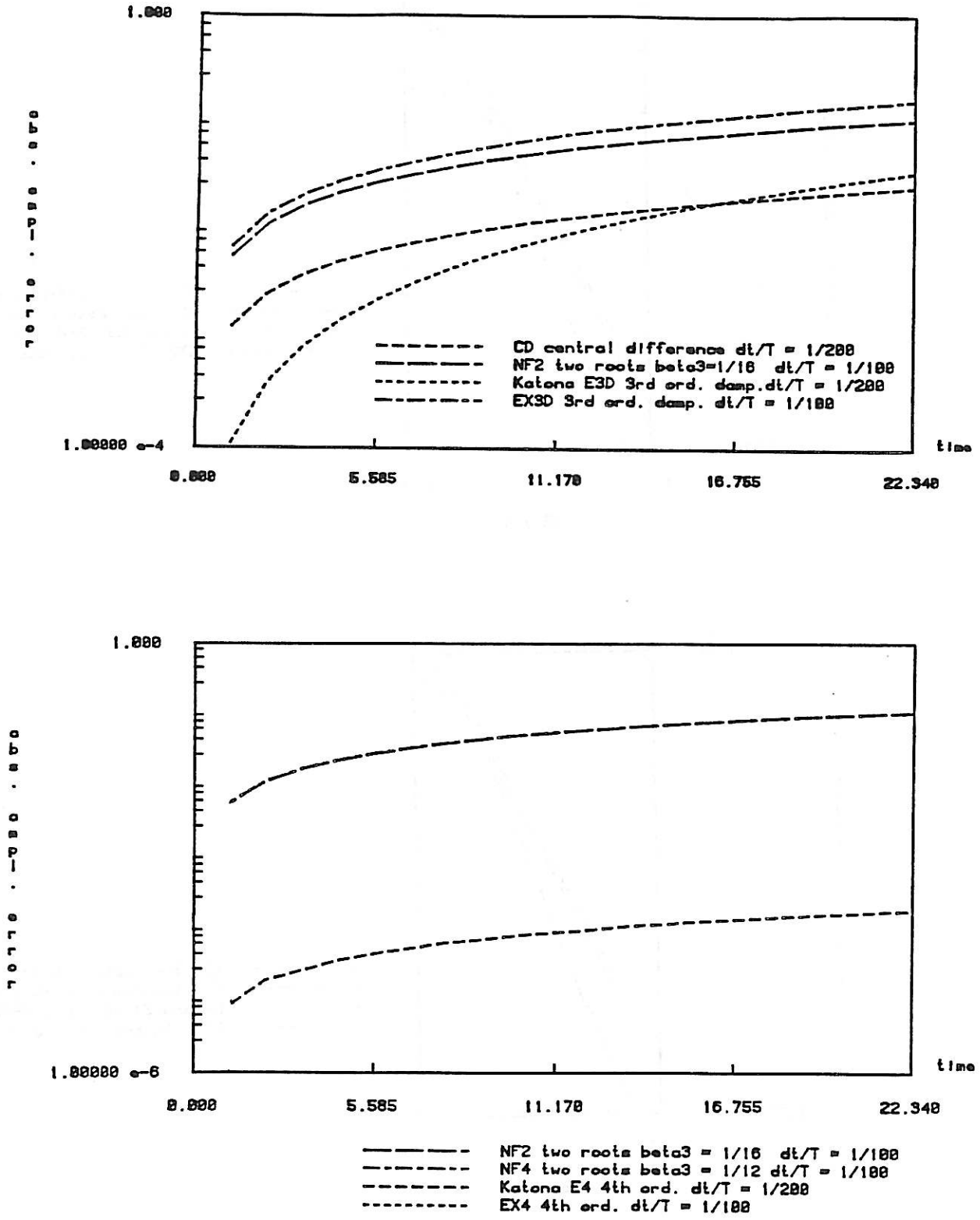


Figure 13 : Absolute error in the displacements at the end of each period over 40 cycles with time steps $\frac{\Delta t}{T} = \frac{1}{100}, \frac{1}{200}$ (free vibration of an undamped softening spring)

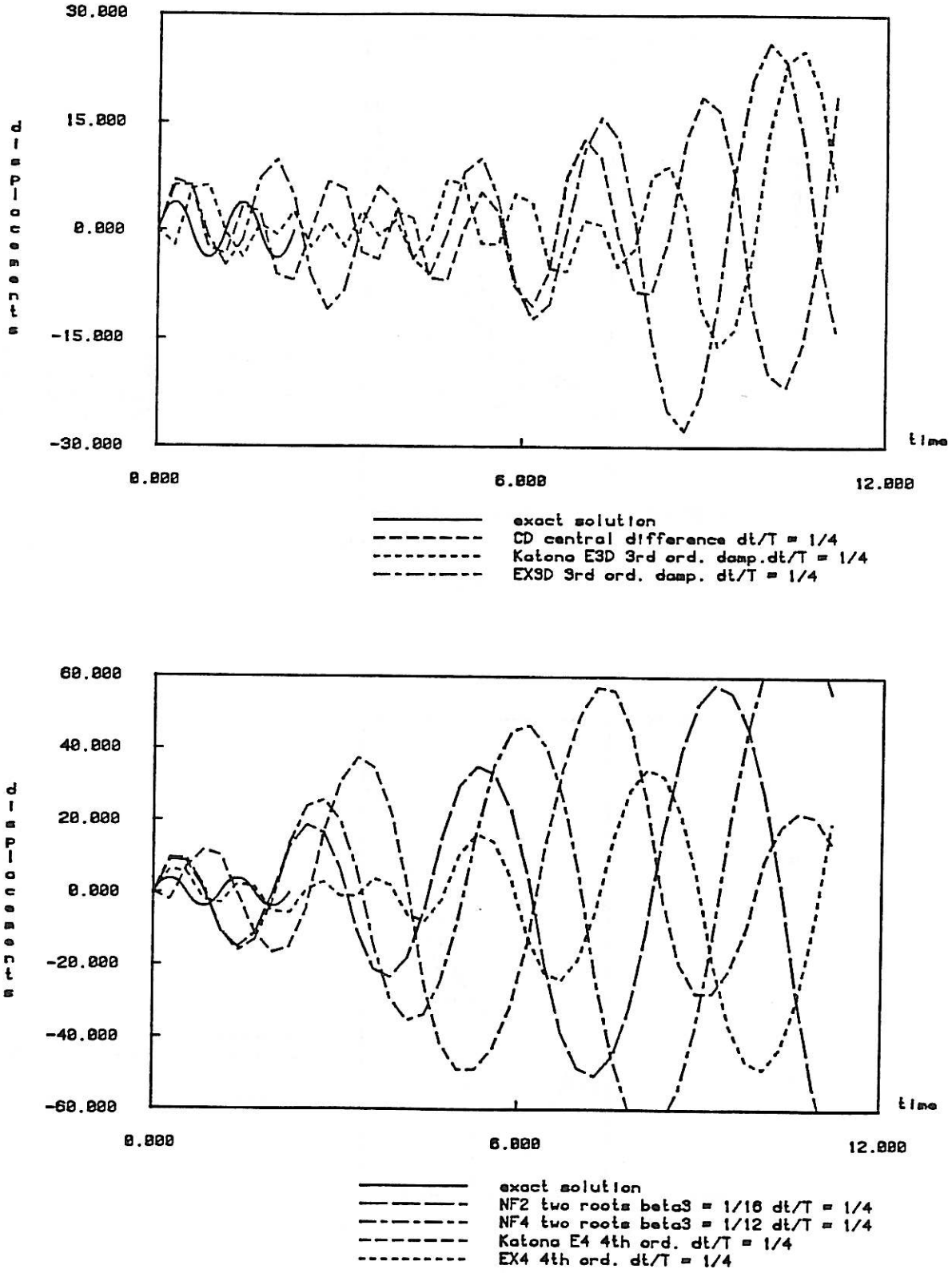


Figure 14 : Initial stability behavior in the displacements during the first 32 time steps $\frac{\Delta t}{T} = \frac{1}{4}$ (free vibration of an undamped softening spring)

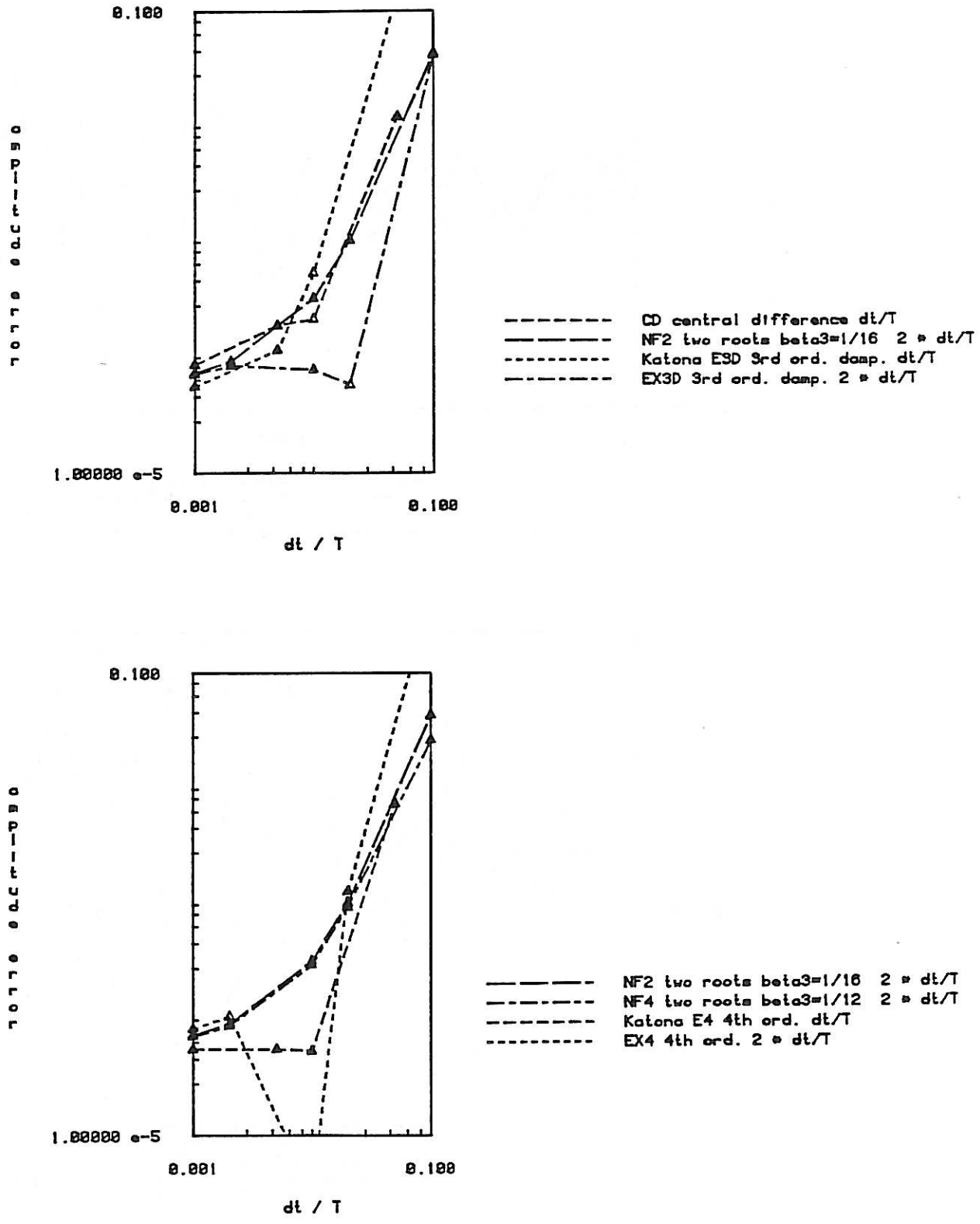
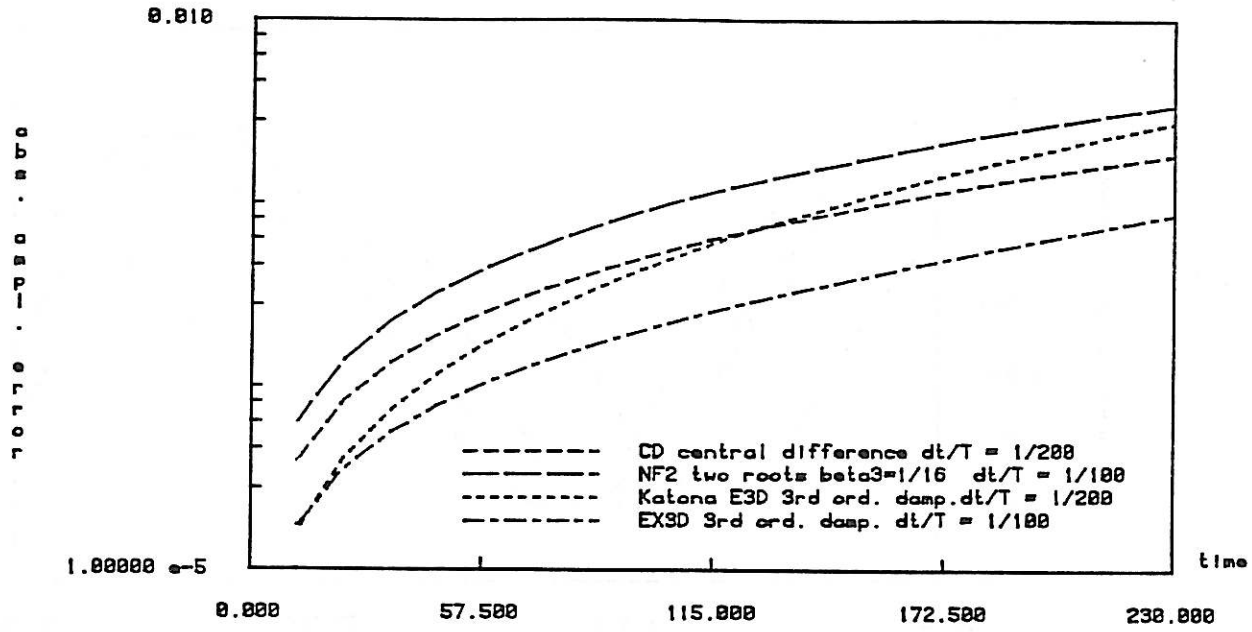


Figure 15 : Convergence in the displacements after 4 periods
(forced vibration of an undamped softening spring tanh u)



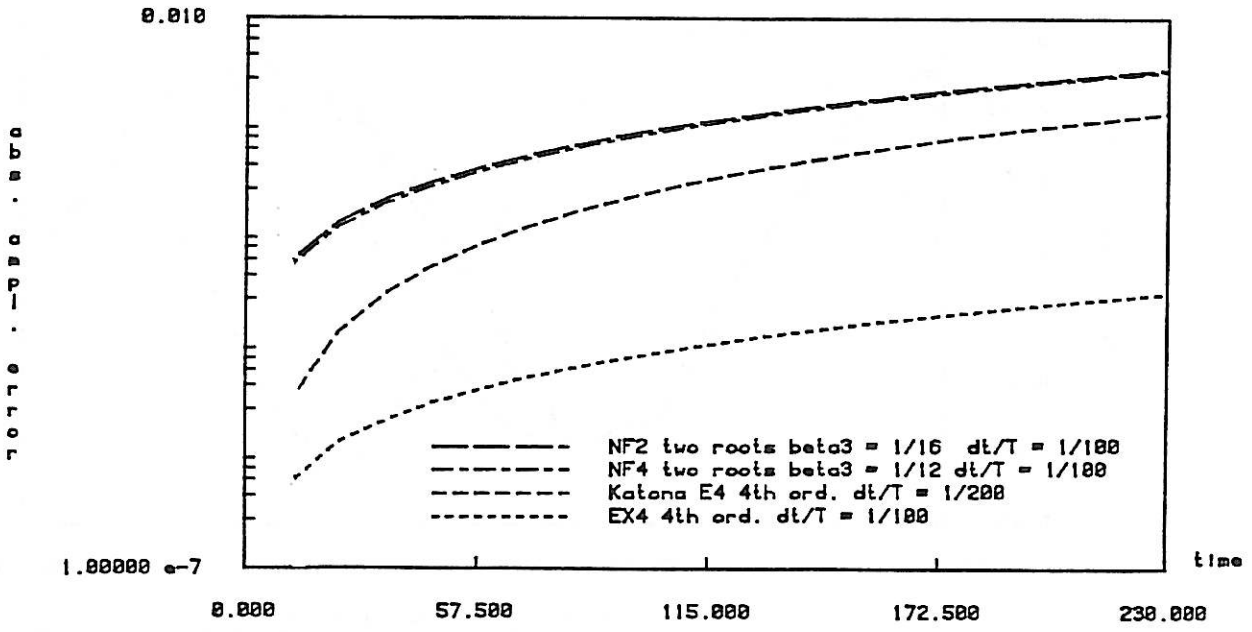
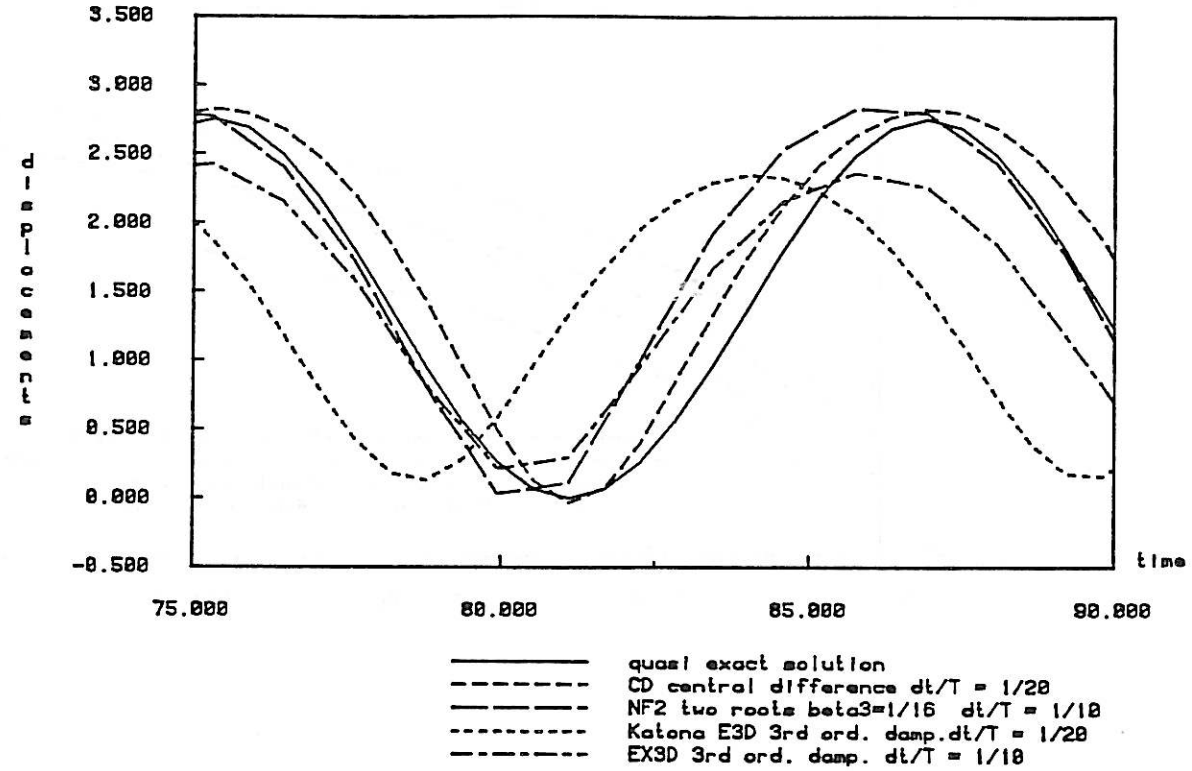


Figure 16 : Absolute error in the displacements at the end of each period over 40 cycles with time steps $\frac{\Delta t}{T} = \frac{1}{100}, \frac{1}{200}$ (forced vibration of an undamped softening spring)



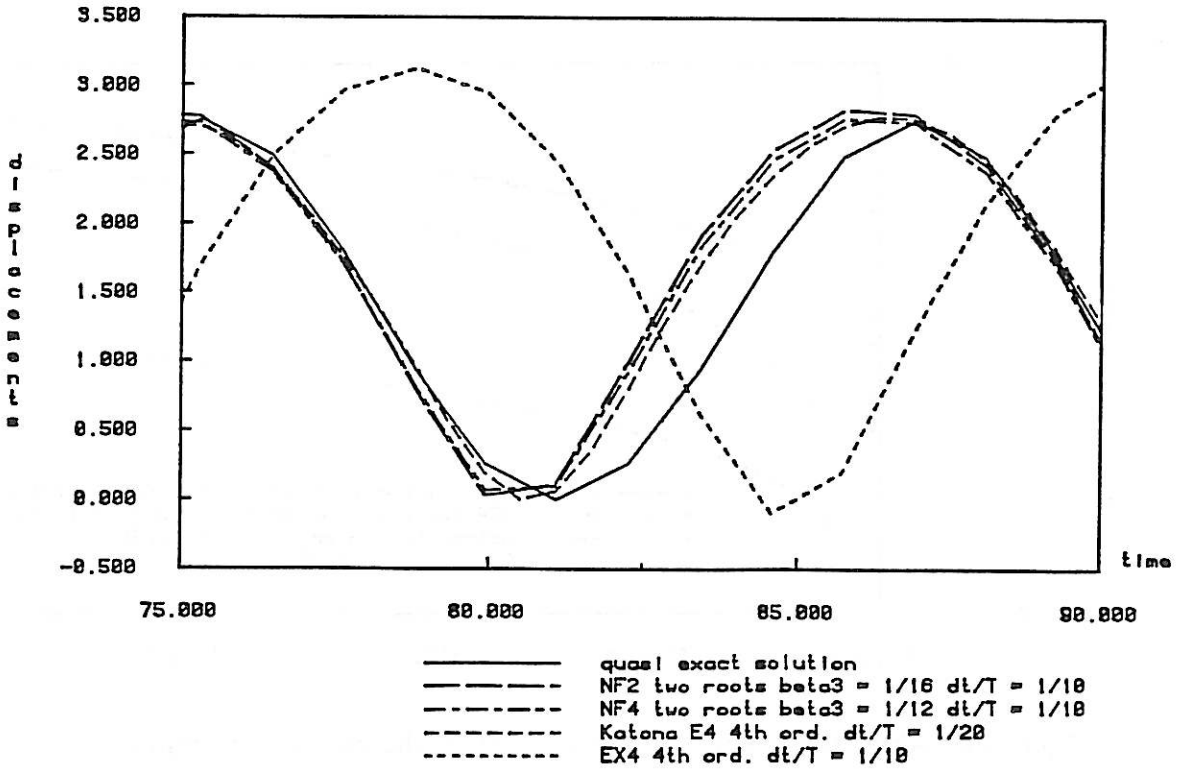


Figure 17 : Displacements after 8 periods with time steps $\frac{\Delta t}{T} = \frac{1}{10}, \frac{1}{20}$

(forced vibration of an undamped softening spring)

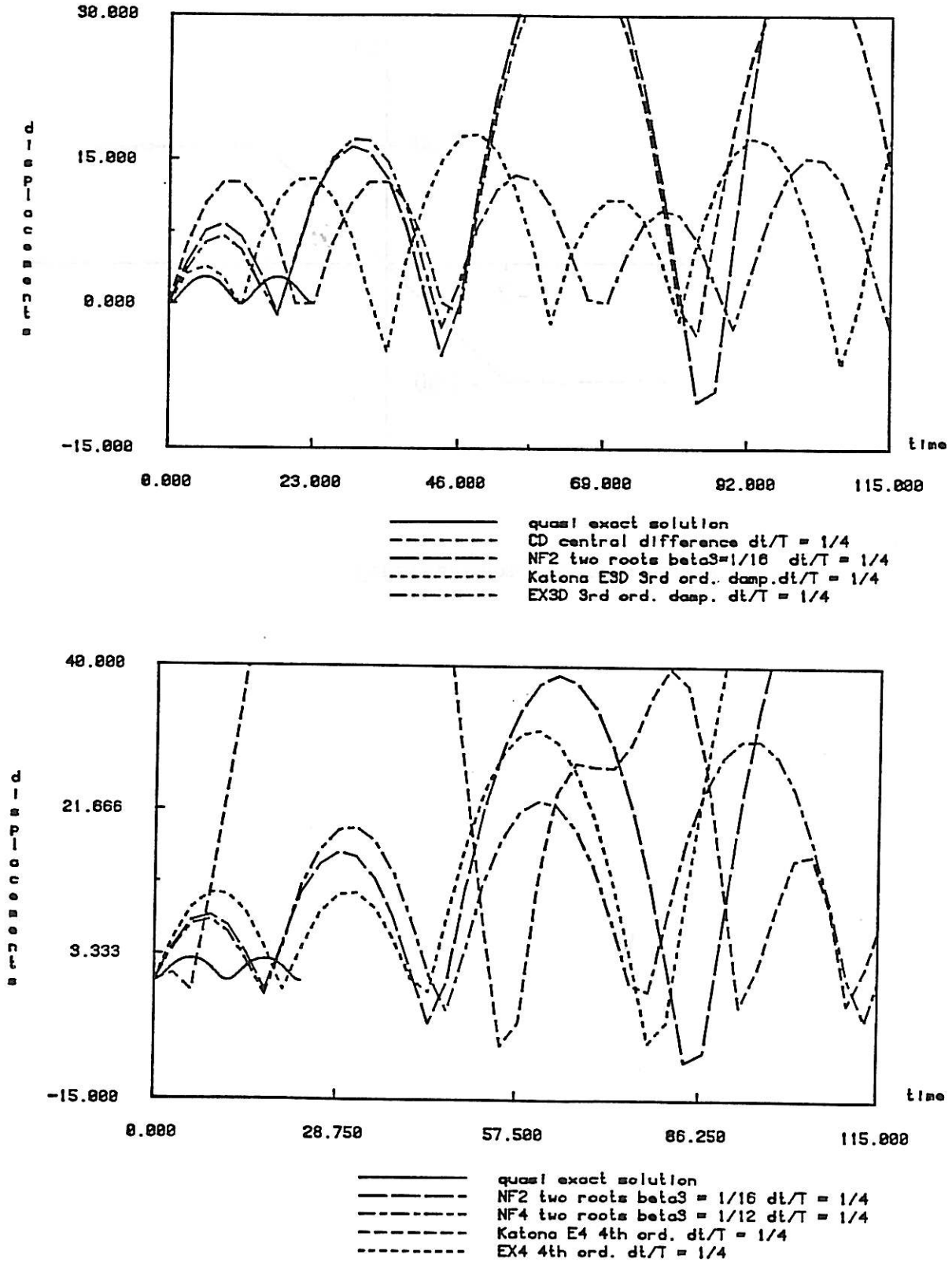


Figure 18 : Initial stability behavior in the displacements during the first 40 time steps $\frac{\Delta t}{T} = \frac{1}{4}$ (forced vibration of an undamped softening spring)

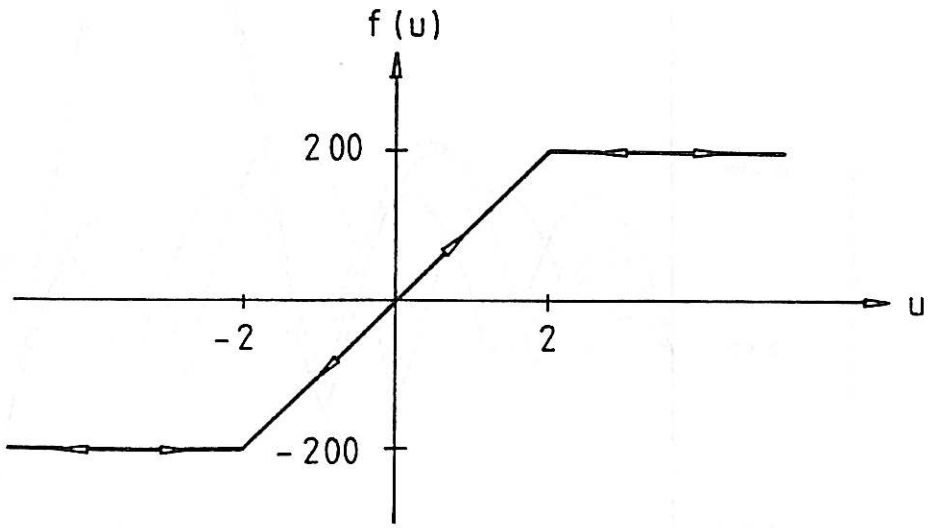


Figure 19 : Bilinear softening spring

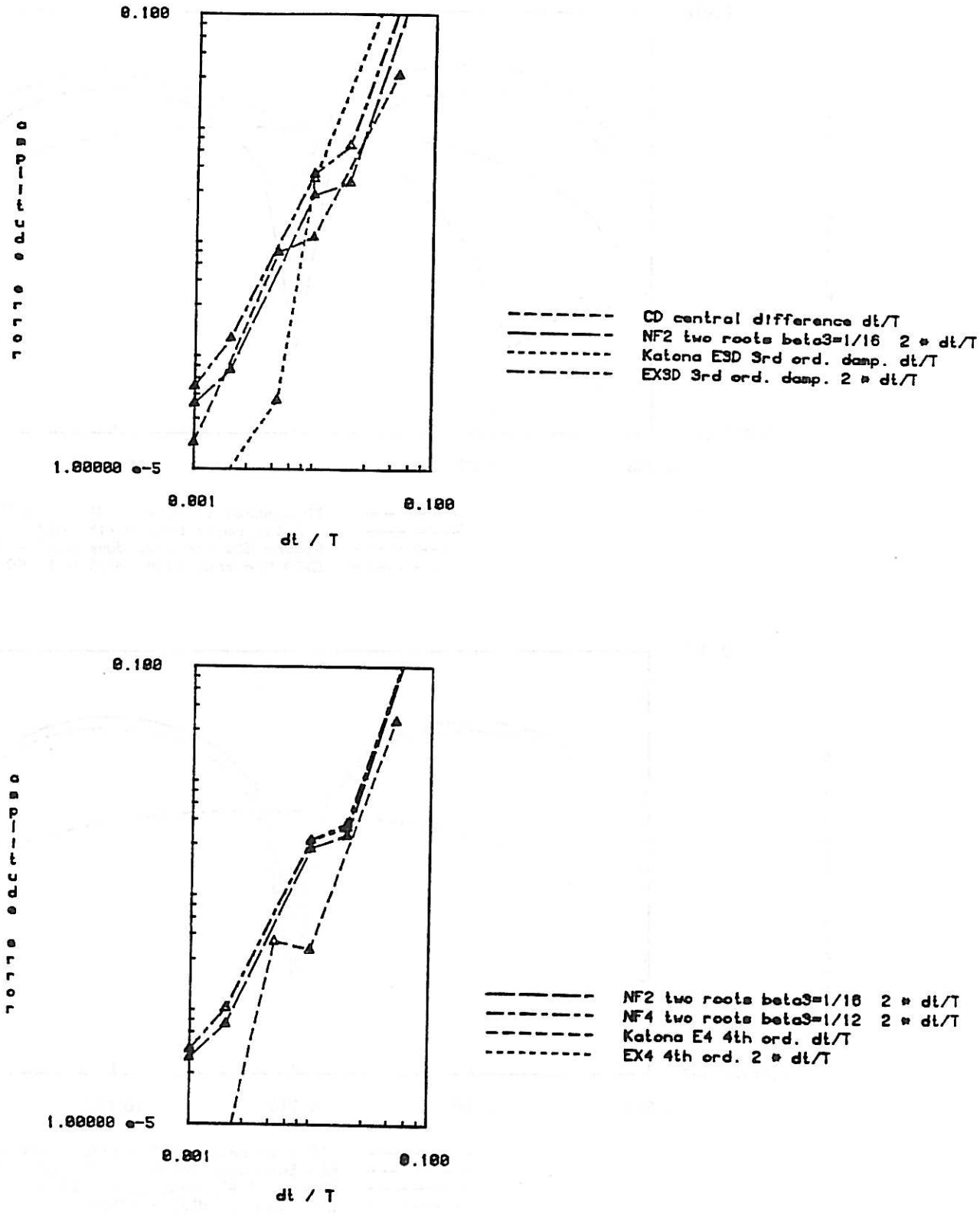
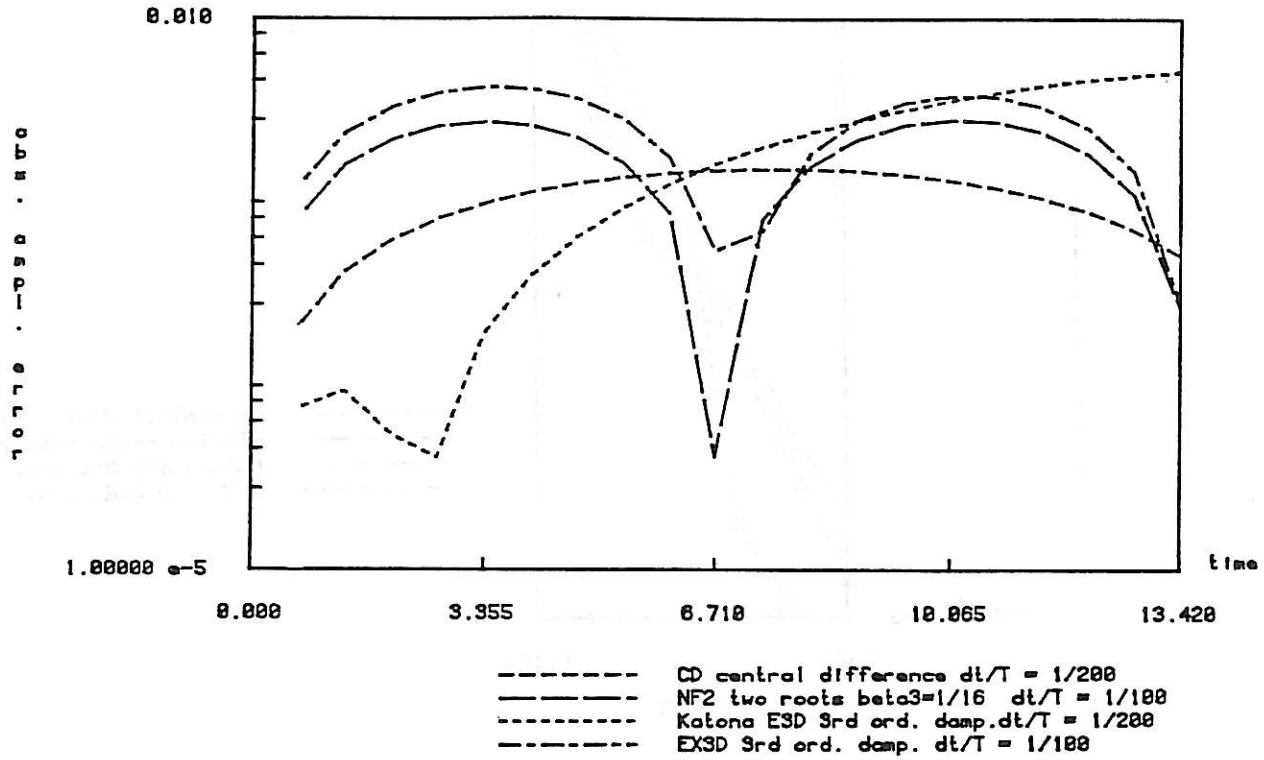


Figure 20 : Convergence in the displacements after 4 periods
(undamped bilinear softening spring)



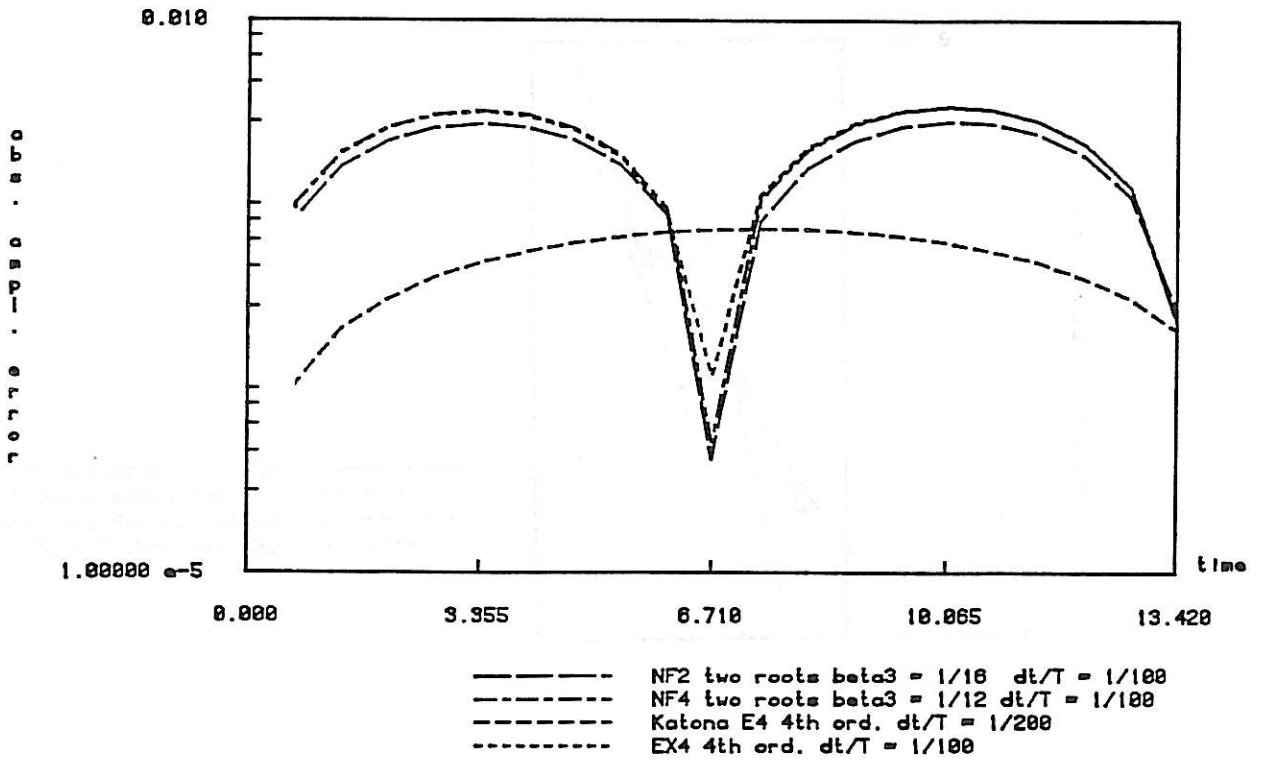


Figure 21 : Absolute error in the displacements at the end of each period over 40 cycles with time steps $\frac{\Delta t}{T} = \frac{1}{100}, \frac{1}{200}$ (undamped bilinear softening spring)

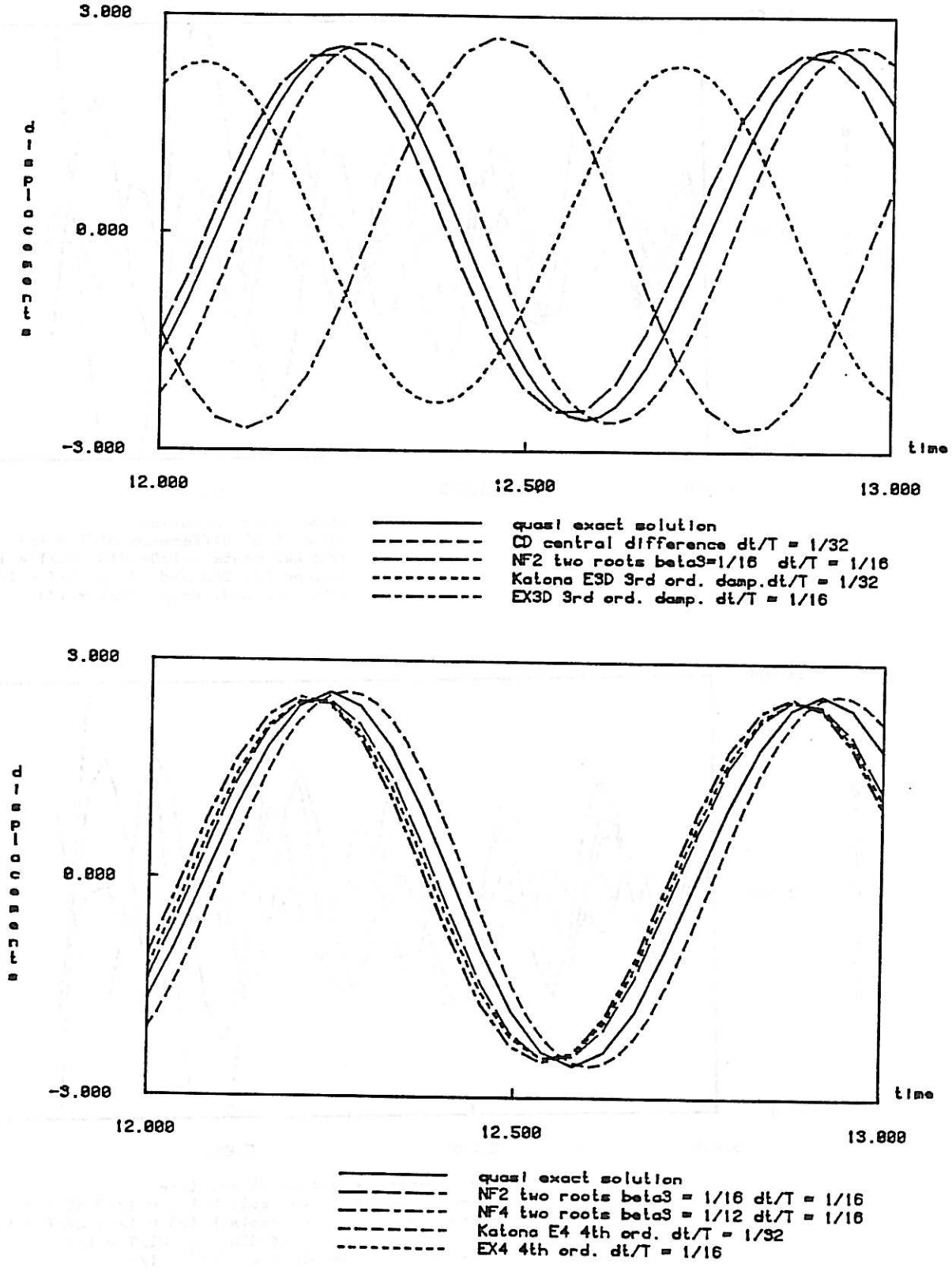


Figure 22 : Displacements after 20 periods with time steps $\frac{\Delta t}{T} = \frac{1}{16}, \frac{1}{32}$

(undamped bilinear softening spring)

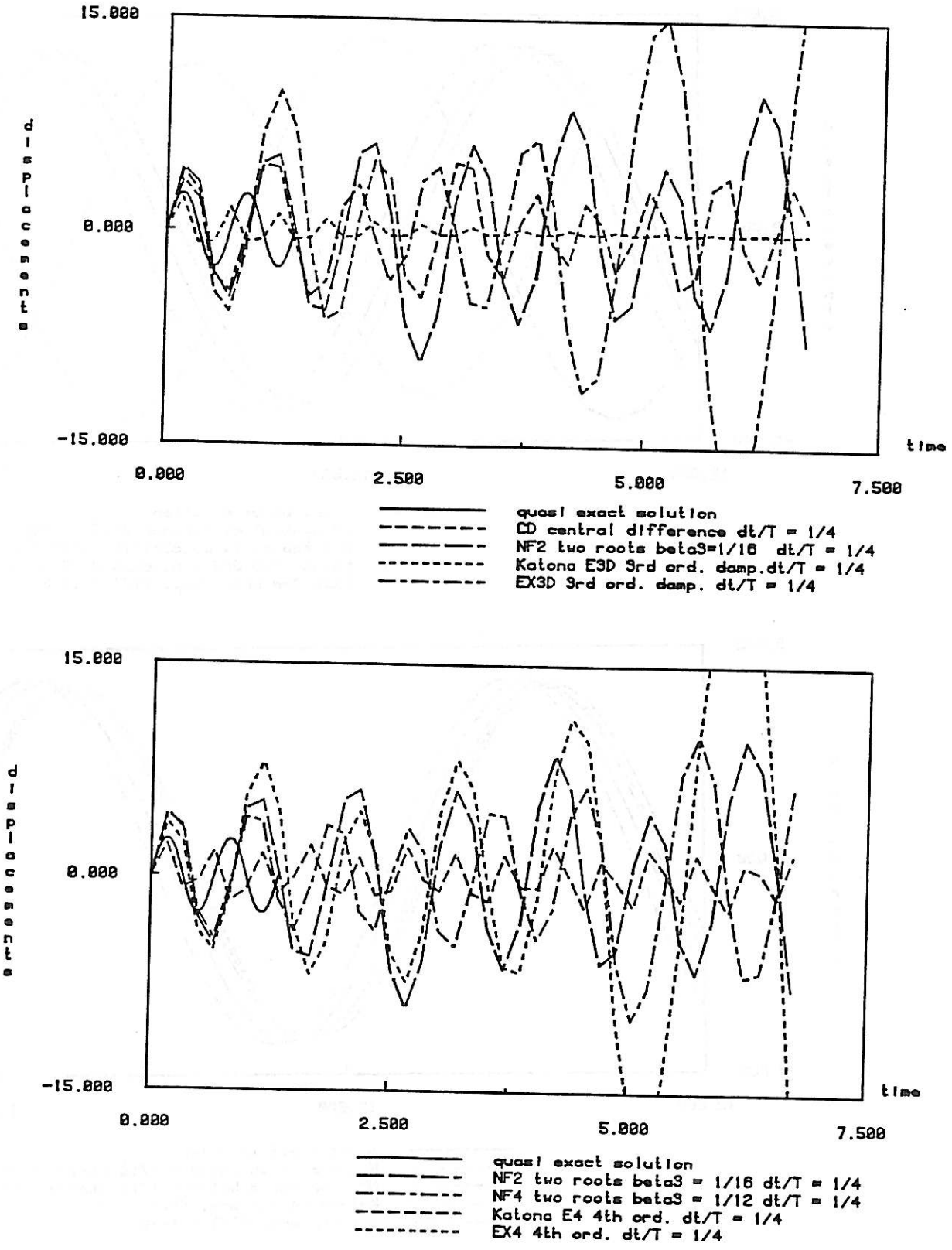


Figure 23 : Initial stability behavior in the displacements during the first 48 time steps $\frac{\Delta t}{T} = \frac{1}{4}$ (undamped bilinear softening spring)