Non-standard, explicit integration algorithms based on linearization for nonlinear dynamic response analysis

J.I. Ramos

E.T.S. Ingenieros Industriales, Universidad de Málaga, Room I-320-D,
Plaza El Ejido, s/n, 29013 Málaga, Spain

Abstract

Non-standard finite difference methods for initial-value problems of second-order ordinary differential equations based on piecewise linearization are developed. Linearization methods provide piecewise analytical solutions which are globally differentiable, result in explicit finite difference algorithms, and are exact for constant coefficients equations with a right-hand side which depends linearly on time. The accuracy of these methods is assessed by obtaining the solutions of several conservative and dissipative, stiff and non-stiff, regular and chaotic problems, and comparing them with those of the MATLAB ode45 solver, state transition matrix algorithms and non-standard Euler techniques.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Linearization methods; Second-order ordinary differential equations; Duffing oscillator; Van der Pol oscillator; Flat potential

1. Introduction

Many numerical algorithms for the efficient and accurate solution of the equations of motion of elastic structures and dynamical problems are now available. In many of these methods, accuracy is often achieved by adapting the time step. Such an adaption is sometimes inconvenient or impossible, such
as, for example, in coupled fluid-structure problems. Moreover, in many circumstances, large time steps may have to be employed [1].

In order to approach the usually opposing views of large time steps and accuracy, some researchers have recurred to classical solutions. For example, Bartels [1] has developed a variable-order method for the numerical integration of initial-value problems (IVP) of ordinary differential equations (odes). This method is based on the state transition matrix and Neumann series and has the advantage of offering an exact solution to a system of ordinary differential equations for arbitrary time steps. The state transition matrix algorithm of Kucharski [2] does not provide a general recursive framework for solving odes whose coefficients or right-hand sides depend on time, whereas the variable-order method of Bartels [1] provides a discrete form of the state transition matrix which is applicable to odes with time-dependent coefficients and right-hand sides, even though the state transition matrix algorithm is only formally valid for linear odes.

Other researchers have dealt with the large step size-accuracy dilemma by means of non-standard finite difference methods, which usually require the development of specific algorithms for each ode but have the advantage that the finite difference discretization is based on the exact solution of the ode or a very good approximation thereof [3–5]. For example, Kojouharov and Welfert [5] have developed non-standard Euler methods based on a physically motivated reduction of a second-order nonlinear ode to a system of two first-order odes and the use of Lie group integrators which have the same linear stability characteristics as the ode and are conservative in the absence of dissipation.

Non-standard finite difference methods which do not require knowledge of the exact solution have been developed by the author and co-workers over a number of years [6–8] for first-order odes. These methods are based on either the full or the partial linearization of nonlinear odes, provide piecewise analytical solutions which are differentiable in the whole time domain, preserve the linear stability properties of the ode, and are exact for constant coefficients odes with a right-hand side that is a linear function of time. However, they do result in cumbersome finite different expressions which usually depend in an exponential manner on the values of the dependent variables at the previous time step. Fortunately, these finite difference expressions can be written in an explicit manner as mappings from one time level to the next and are, therefore, explicit, and their solution can be usually obtained quite fast. These linearization methods were initially developed for systems of first-order nonlinear odes [6–8].

In this paper, we present non-standard finite difference methods based on the piecewise linearization for nonlinear second-order odes, which provide exact solutions (in the absence of round-off errors) for constant coefficients odes whose right-hand sides depend linearly on time, and which yield non-standard, explicit, nonlinear formulae for determining the values of the dis-
placement and velocity of one-degree of freedom systems at time level \((n + 1)\) in terms of these variables at time level \(n\). Moreover, these methods can be used in an adaptive manner because their finite difference expressions depend on the linearization of the nonlinear terms with respect to the displacement, velocity and time. In this paper, we shall only consider linearization methods with fixed time step, but provide comparisons with the results from techniques based on the state transition matrix [1], non-standard algorithms [5] and the variable-order MATLAB ode45 solver, and assess the effects of different time steps on the displacement, velocity, phase diagram and energy conservation for both conservative and dissipative, stiff and non-stiff, regular and chaotic systems.

The paper has been organized as follows. In Section 2, linearization methods for second-order, nonlinear odes are presented, while, in Section 3, these methods are applied to a variety of stiff and non-stiff, conservative and dissipative nonlinear odes, and the results are compared with those obtained with the MATLAB solver ode45 and state transition matrix and non-standard algorithms. A summary of the main conclusions puts an end to the paper.

2. Linearization methods

Consider the following second-order nonlinear ode

\[
\ddot{u} = f(t, u, \dot{u}), \quad 0 < t \leq T, \quad (1)
\]

subject to

\[
u(0) = u_0, \quad \dot{u}(0) = \dot{u}_0, \quad (2)
\]

where the dots denote differentiation with respect to time \((t)\), \(u_0\) and \(\dot{u}_0\) are known, and \(f(t, u, \dot{u})\) is a nonlinear function of \(t\), \(u\) and \(\dot{u}\), which is regular at \(t = 0\). Consider the interval \((0, T]\) and divide it into a series of non-overlapping subintervals \((t_n, t_{n+1}]\) such that \(t_0 = 0\). In each subinterval, \(f(t, u, \dot{u})\) may be linearized as follows. If \(f(t, u, \dot{u})\) is regular, \(f\) may be approximated by the first terms of its Taylor series expansion around \((t_n, u_n, \dot{u}_n)\), so that Eq. (1) may be approximated by

\[
\ddot{u} = f_n + J_n(\dot{u} - \dot{u}_n) + H_n(u - u_n) + T_n(t - t_n), \quad t_n < t \leq t_{n+1}, \quad (3)
\]

subject to

\[
u(t_n) = u_n, \quad \dot{u}(t_n) = \dot{u}_n, \quad (4)
\]

where \(H_n = \frac{\partial^2 f}{\partial u^2}(t_n, u_n, \dot{u}_n), \quad J_n = \frac{\partial f}{\partial \dot{u}}(t_n, u_n, \dot{u}_n)\) and \(T_n = \frac{\partial f}{\partial t}(t_n, u_n, \dot{u}_n)\).

Eq. (3) is a second-order linear ode whose analytical solution may be easily determined as a function of \(J_n, H_n, T_n, u_n\) and \(\dot{u}_n\). For example, if \(R_n = (J_n/2)^2 + H_n > 0\) and \(H_n \neq 0\), the solution of Eq. (3) can be written as
\[ u(t) = A_n \exp(\lambda_n^+(t - t_n)) + B_n \exp(\lambda_n^-(t - t_n)) + \zeta_n + \beta_n(t - t_n), \quad t_n \leq t \leq t_{n+1}, \] (5)

where \( \lambda_n^\pm = J_n/2 \pm \sqrt{R_n} \), \( \beta_n = -J_n/H_n \), \( \zeta_n = -(Q_n + J_n\beta_n)/H_n \), \( Q_n = f_n - J_n\dot{u}_n - H_n u_n \), and \( A_n \) and \( B_n \) are constants which can be easily determined from Eq. (4) as

\[ A_n = (\dot{u}_n - \beta_n - \lambda_n^- (u_n - \zeta_n))/((\lambda_n^+ - \lambda_n^-)), \] (6)

\[ B_n = (\lambda_n^+(u_n - \zeta_n) - \dot{u}_n + \beta_n)/((\lambda_n^+ - \lambda_n^-)), \] (7)

and, therefore, Eq. (5) implies that

\[ u_{n+1} = A_n \exp(\lambda_n^+ \Delta t_n) + B_n \exp(\lambda_n^- \Delta t_n) + \zeta_n + \beta_n \Delta t_n, \] (8)

\[ \dot{u}_{n+1} = \lambda_n^+ A_n \exp(\lambda_n^+ \Delta t_n) + \lambda_n^- B_n \exp(\lambda_n^- \Delta t_n) + \beta_n, \] (9)

where \( \Delta t_n = t_{n+1} - t_n \) is the time step which may be varied according to the evolution of the solution, i.e., according to the values of \( \lambda_n^+ \) and \( \lambda_n^- \) which depend on the variation of \( f \) with respect to \( u \) and \( \dot{u} \).

Eqs. (8) and (9) are exact for functions \( f(t, u, \dot{u}) \) which are linear functions of their arguments, and correspond to explicit, albeit cumbersome, expressions for \( u_{n+1} \) and \( \dot{u}_{n+1} \) in terms of the values of \( u_n \) and \( \dot{u}_n \), and result in a non-standard finite difference method. Moreover, the piecewise analytical solutions provided by Eq. (5) are continuous with continuous derivatives, cf. Eqs. (3) and (4), for \( t \geq 0 \).

Analytical solutions to Eq. (3) can also be obtained for other cases different from the one considered above, e.g., \( R_n \leq 0 \), and these solutions provide explicit maps for \( u_{n+1} \) and \( \dot{u}_{n+1} \) as functions of \( u_n \) and \( \dot{u}_n \); these solutions are not presented here.

The approximation of Eq. (1) by Eq. (3) is here referred to as full linearization because the nonlinear function \( f(t, u, \dot{u}) \) has been linearized with respect to all its arguments. Partial linearization methods correspond to the linearization of \( f \) with respect to some (but not all) of its arguments, whereas frozen methods correspond to the approximation of \( f \) by \( f_n \). An example of partial linearization corresponds to the approximation of Eq. (1) by

\[ \dot{u} = f_n + J_n(\dot{u} - \dot{u}_n), \quad t_n < t \leq t_{n+1}, \] (10)

where it has been implicitly assumed that the term \( H_n(u - u_n) + T_n(t - t_n) \) is much smaller than the terms that appear in the right-hand side of Eq. (10).

Frozen methods correspond to the approximation

\[ \dot{u} = f_n, \quad t_n < t \leq t_{n+1}, \] (11)

whose analytical solution can be written as

\[ u = f_n(t - t_n)^2/2 + \zeta_n + \beta_n(t - t_n), \quad t_n < t \leq t_{n+1}, \] (12)
where \( \alpha_n = u_n \) and \( \beta_n = \dot{u}_{n+1} \) and, therefore,

\[
 u_{n+1} = f_n \Delta t_n^2/2 + \alpha_n + \beta_n \Delta t_n, \quad \dot{u}_{n+1} = f_n \Delta t_n + \beta_n, \tag{13}
\]
i.e., they correspond to Taylor’s (explicit) second-order accurate method.

The nonlinear mappings or difference equations corresponding to Eqs. (8) and (9) have been derived by considering that \( f \) is a function of three variables \( t, u \) and \( \dot{u} \). Moreover, since these equations have been obtained by approximating \( f \) by the first (linear) terms of its Taylor series expansion, while second- and higher-order terms have been neglected, the local step size \( \Delta t_n \) may be determined in an adaptive manner from the condition that the second-order terms be much smaller than the first-order ones, or from the condition that \( u_{n+1} \) and \( \dot{u}_{n+1} \) do not differ significantly from \( u_n \) and \( \dot{u}_n \), respectively. In this paper, only the results of linearization methods with fixed time steps are reported.

3. Presentation and discussion of results

In order to assess the advantages and limitations and the accuracy of linearization methods, we have applied them to a variety of IVP in odes arising in nonlinear dynamics as indicated in the following examples. In these examples, we show the discrete displacement and velocity, i.e., \( u_n \) and \( \dot{u}_n \), respectively, and the discrete energies \( E_n \) at different (discrete) times, \( t_n \), and the phase diagrams corresponding to the values of \( u_n \) and \( \dot{u}_n \), rather than the displacement, velocity, energy and phase diagrams which result from Eq. (5). In addition, the graphs presented here have been obtained by joining by means of straight lines successive discrete points and, therefore, may exhibit cusps which would be absent if the solution corresponding to Eq. (5) had been drawn.

**Example 1.** This example corresponds to the dynamic response of a single degree of freedom spring-mass system where the stiffness of the spring is a function of time and whose dynamic equation can be written as [1]

\[
 \ddot{x} = -kx = -(a_1 t + a_2)^{-4} x, \quad x(0) = 0, \quad \dot{x}(0) = 1, \tag{14}
\]
where \( a_1 = 0.1 \) and \( a_2 = 100^{-1/4} \).

Eq. (14) indicates that the stiffness decreases as time increases; \( k = 100 \) at \( t = 0 \) and \( k \) is nearly zero at \( t = 50 \). This large variation in stiffness presents a challenging problem for most numerical solvers because the frequency varies dramatically over a moderate time scale and, usually, this problem is solved numerically by means of variable-order and/or variable-step methods. Here, we have used a fixed time step and some sample results are presented in Fig. 1. This figure was obtained with a full linearization technique and indicates that there are very few differences between the results corresponding to \( \Delta t = 0.001 \).
and $\Delta t = 0.01$; the results corresponding to $\Delta t = 0.001$ are indistinguishable from those obtained with the MATLAB variable-order ode45 solver when a tolerance of $10^{-13}$ is employed. The MATLAB ode45 solver required about 20,000 time steps to obtain the solution up to $T = 50$, i.e., an average step size equal to 0.0025. Fig. 1 also shows that the accuracy of full linearization methods degrades as the time step is increased; for $\Delta t = 0.1$ and 0.2, the results presented in Fig. 1 indicate that the displacement, for $t$ greater than approximately 30, is larger than that corresponding to $\Delta t = 0.001$. Fig. 1 also indicates that the velocity increases as the time step is increased for times larger than about 21.

Fig. 2 illustrates the results of the full linearization method for $t \leq 5$ and indicates that large time steps predicts lower maxima and less shallow minima than those corresponding to $\Delta t = 0.001$ and $\Delta t = 0.01$ and the MATLAB
variable-order ode45 solver. This figure also shows that the phase differences in the displacement and velocity increase as the time step is increased.

The results presented in Figs. 1 and 2 indicate that the full linearization method presented in this paper with $\Delta t = 0.01$ produces nearly identical results to those of the MATLAB variable-order ode45 solver which requires an average time step equal to about 0.0025. Furthermore, since the full linearization method yields explicit expressions for the displacement and velocity, it is more efficient than the MATLAB ode45 solver.

It is interesting to point that Bartels [1] obtained the solution of Eq. (14) with an O(V) variable-order state transition matrix algorithm and a tolerance of $10^{-12}$ and obtained good agreement with the MATLAB variable-order ode15 solver and a tolerance of $10^{-13}$, when the former employed step sizes equal to 0.1, 0.2 and 0.4. Therefore, for the same accuracy, the full linearization method presented in this paper requires smaller time steps than the variable-order state transition algorithm.

Fig. 3 shows the numerical results obtained with the full linearization method presented in this paper (solid line), partial linearization with respect to $x$ (dashed line), the linearization

$$\ddot{x} = f_{n+1} + J_{n+1}(x - x_n) + T_{n+1}(t - t_{n+1}), \quad t_n < t \leq t_{n+1}$$  \hspace{1cm} (15)

(dashed-dotted line) where $f_{n+1} = f(t_{n+1}, x_n)$ and the frozen technique (dotted line). The results presented in Fig. 3 indicate that, for $\Delta t = 0.01$, there are very few differences between the results corresponding to the full linearization method presented in this paper and the results of Eq. (15); the results of the partial linearization with respect to $x$ are comparable to, but lower than those corresponding to the full linearization scheme, whereas the frozen technique predicts higher relative maxima and shallower relative minima than the full

---

Fig. 3. Example 1: displacement (left) and velocity (right) as functions of time. $\Delta t = 0.01$. $x(0) = 0$, $\dot{x}(0) = 1$. Solid line: full linearization; dashed line: (partial) linearization with respect to $x$; dashed-dotted line: linearization as per Eq. (15); dotted line: frozen method as per Eq. (11).
linearization method. The frozen method underpredicts the displacement and velocity for times larger than about 26.5 and 17, respectively.

The fixed time step required by fully linearization methods is about one half of that required by the MATLAB ode45 solver and smaller than the O(V) variable-order state transition matrix method of Bartels to achieve the same accuracy. However, this is somewhat compensated by the fact that the full linearization method provides differentiable solutions and explicit expressions for \( u_n \) and \( \dot{u}_n \) (cf. Eqs. (8) and (9)) which result in a fast numerical technique.

**Example 2.** This example corresponds to

\[
\ddot{x} = -c x - d \dot{x} - a x^m + g \cos bt,
\]

where \( m = 3, a = b = 1, g = 0.3, d = 0.15, x(0) = -1, \dot{x} = 1 \) and is referred to as Duffing’s equation. This equation was solved by means of the full linearization method presented in this paper with a fixed time step and the MATLAB variable-order ode45 solver with a tolerance of \( 10^{-7} \). This solver required about 4000 time steps for \( T = 50 \) with an average time step of about 0.0125.

Fig. 4 shows that the accuracy of the full linearization method degrades as the time step is increased and only the results corresponding to \( \Delta t = 10^{-6} \) were found to be indistinguishable from those of the MATLAB ode45 solver and a tolerance of \( 10^{-10} \). Fig. 4 also shows that the deviation from the solution obtained with \( \Delta t = 10^{-6} \) appears at lower times as the time step is increased. For example, the discrepancies between the solutions obtained with \( \Delta t = 10^{-5} \) and \( \Delta t = 10^{-4} \) are noticeable after about \( t = 37 \) and 31, respectively; and those between \( \Delta t = 10^{-6} \) and \( \Delta t = 10^{-3} \) are noticeable after about \( t = 19 \).

Although not shown here, the differences between the results obtained with the full linearization method and those of a partial linearization with respect to \( x \) and \( \dot{x} \) were found to be small, whereas those between the full linearization method and a frozen technique were very large even for \( \Delta t = 10^{-6} \).

---

**Fig. 4.** Example 2: displacement (left) and velocity (right) as functions of time. \( x(0) = -1, \dot{x}(0) = 1 \). Solid line: \( \Delta t = 10^{-6} \); dashed line: \( \Delta t = 10^{-5} \); dashed-dotted line: \( \Delta t = 10^{-4} \); dotted line: \( \Delta t = 10^{-3} \).
Bartels [1] obtained the solution of Eq. (16) with his O(O), O(I), O(II), O(III) and O(IV) variable-order state transition matrix algorithms, a time step equal to 0.05 and a tolerance of $10^{-7}$, but only obtained good agreement with the solution of the MATLAB variable-order ode15 solver with the O(IV) variable-order method. The numerical solutions of his O(III) method are in close agreement with the full linearization employed in this paper with $\Delta t = 10^{-5}$, but the solutions of his O(O), O(I) and O(III) variable-order state transition matrix algorithms differ very much from those of the MATLAB ode15 solver and from the linearization technique with $\Delta t = 10^{-3}$ and $10^{-4}$. Therefore, for the same accuracy, the full linearization method requires smaller time steps than the MATLAB ode15 solver which, in turn, requires smaller time steps than the variable-order state transition matrix algorithms.

**Example 3.** This example corresponds to the linear damped oscillator governed by

$$\ddot{x} + \epsilon \dot{x} + \omega^2 x = 0, \quad (17)$$

with $x(0) = 3$, $\dot{x}(0) = 0$, $\omega = 4$, and $T = 20$. For $\epsilon = 0$, the total energy $e = \dot{x}^2/2 + \omega^2 x^2/2$ is conserved.

Eq. (17) was solved with the full linearization method presented in this paper and the MATLAB ode45 solver with a tolerance of $10^{-10}$, and the former gave results indistinguishable from the latter for $\Delta t = 0.001$ for both $\epsilon = 1$ and 0.

Fig. 5 illustrates the results obtained with the full linearization method for $\epsilon = 1$ and indicates small differences between the results obtained for a variety of fixed time steps, especially in $x$, $\dot{x}$ and $E = e/e(0)$; the latter decreases with time.

Since this example is linear, the full linearization presented in this paper provides the exact solution in exact arithmetic. The small differences in the displacement and velocity profiles shown in Fig. 5 are a consequence of the fact that these profiles do not correspond to Eq. (5); they do correspond to the discrete values of $(t_n, x_n)$ and $(t_n, \dot{x}_n)$ and the linear interpolation of these values employed in the displacement and velocity diagrams. A similar comment applies to the phase diagram that corresponds to the linear interpolation of the discrete coordinates $(x_n, \dot{x}_n)$. As a consequence of this linear interpolation, the phase diagram shows cusps, especially for large time steps, but the cusp points have the same coordinates regardless of the time step.

Quite different results are obtained for $\epsilon = 0$ as the results presented in Fig. 6 show. Since, for $\epsilon = 0$, the system is conservative with only a frequency, the phase plane should be a simple closed curve. The numerical results indicate that the phase plane is indeed a closed curve regardless of the time step because the full linearization method provides the exact solution to Eq. (17) in exact arithmetic, even though the displacement, velocity and phase diagrams presented in Fig. 6 show small discrepancies which are caused by the linear
interpolation used to represent the displacement, velocity and phase diagrams. Fig. 6 also shows that the full linearization method almost conserves exactly the energy for $\Delta t = 0.001$, and that larger errors in energy conservation occur for $\Delta t = 0.4$ but are small; these errors are associated with the round-off errors incurred in Eqs. (8) and (9). Although not shown here, the MATLAB ode45 solver with a tolerance of $10^{-10}$ predicts a quasi-linear departure of the energy from its initial value for $\epsilon = 0$.

The results presented in Fig. 6 indicate that, for Hamiltonian systems, the conservation of energy is not a good indicator of the accuracy of the numerical solution and that attention should be placed on the phase diagram.

Kojouharov and Welfert [5] solved Eq. (17) by means of a non-standard difference Euler method with a step size equal to 0.1, obtained results in agreement with those of the MATLAB ode45 solver with a tolerance of $10^{-10}$ for $\epsilon = 1$ and 0, and found that their non-standard Euler method captures the dissipation rate exactly, but it increases the frequency of oscillations compared with the exact solution for $\epsilon = 1$. Therefore, for the same accuracy, the full
The linearization method requires smaller time steps than the non-standard difference Euler method.

Example 4. This example corresponds to the Duffing oscillator governed by

\[
\ddot{x} + f(x)x = 0,
\]

where \( f(x) = \omega^2 (1 + x^2/\lambda^2) \) with \( x(0) = 1, \dot{x}(0) = 0, \omega = \lambda = 1 \) and \( T = 50 \). The total energy \( e = \dot{x}^2/2 + \omega^2(x^2/2 + x^4/(4\lambda^2)) \) is conserved.

Fig. 7 shows that the accuracy of the full linearization method decreases as the time step is increased. In particular, \( \Delta t = 0.001, 0.01 \) and 0.1 result in constant amplitude oscillations, a closed phase diagram characterized by a single closed curve, and almost energy conservation; however, a time step equal to 0.4 results in damped oscillations and shrinking phase diagrams. It must be pointed out that the full linearization method presented in this paper with \( \Delta t = 0.001 \) yielded indistinguishable results from those of the MATLAB ode45 solver with a tolerance of \( 10^{-10} \). The MATLAB solver predicts that the total energy

\[
e = \dot{x}^2/2 + \omega^2(x^2/2 + x^4/(4\lambda^2))
\]

is conserved.
Energy deviates from its initial value in a quasi-linear manner as a function of time, whereas a non-standard Euler method preserves the total energy over large intervals of time even for a time step equal to 0.5 [5].

Example 5. This example corresponds to the Van der Pol oscillator governed by

\[ \ddot{x} + g(x)\dot{x} + \omega^2 x = 0, \]  

where \( g(x) = \epsilon(x^2/\mu^2 - 1) \) with \( x(0) = 3, \dot{x}(0) = 0, \omega = \lambda = 1 \), \( \epsilon = 2 \) and \( T = 50 \). In this case, the energy \( e = \dot{x}^2/2 + \omega^2x^2 \) is not conserved.

The results presented in Fig. 8 show that the full linearization method predicts accurately the displacement, velocity, phase diagram and energy for \( \Delta t = 0.1, 0.01, 0.001 \) and 0.0001, and that the major differences between these time steps are observed in the phase diagram, but are small. Moreover, the full linearization method with \( \Delta t = 0.0001 \) provided results indistinguishable from those of the MATLAB ode45 solver with a tolerance of \( 10^{-10} \). By way of
In contrast, the non-standard Euler method employed by Kojouharov and Welfert [5] shows the right behaviour of both $x$ and $\dot{x}$ but the wrong phase for a step size equal to 0.1 compared with the solution provided by the MATLAB ode45 solver. This error in phase can be corrected by modifying $\dot{x}$. Therefore, for the same accuracy, the full linearization method requires about the same time steps as the non-standard difference Euler method.

**Example 6.** This example corresponds to a flat potential governed by

$$\ddot{x} + f(x)x = 0,$$

where $f(x) = x^2/(1 + x^2)$ with $x(0) = 2$, $\dot{x}(0) = 0$ and $T = 50$. The total energy $e = \dot{x}^2/2 + x^2/2 - [\ln(1 + x^2)]/2$ is conserved, and the potential energy is relatively flat around $x = 0$.

Fig. 9 indicates that the full linearization method predicts accurately the dynamics for $\Delta t = 0.2, 0.1, 0.01$ and 0.001, even though small differences between the results corresponding to $\Delta t = 0.2$ and 0.001 can be observed in the
displacement and energy. The results of the full linearization method presented in this paper with $\Delta t = 0.001$ were found to be indistinguishable from those of the MATLAB ode45 solver with a tolerance of $10^{-10}$. The non-standard difference Euler method of Kojouharov and Welfert [5] requires a step size equal to 0.01 to achieve results in agreement with those of the MATLAB ode45 solver; however, this method is iterative, requires the solution of a nonlinear equation, and preserves the initial energy with an error which is comparable to the accuracy with which this nonlinear equation is solved. In addition, the non-standard difference Euler method of Kojouharov and Welfert [5] results in an ill-conditioned problem caused by stiffness.

Example 7. This example corresponds to a quantum potential governed by

$$\ddot{x} + g(x)\dot{x} + f(x)x = 0,$$

where $f(x) = \omega^2(x^2/\lambda^2 - 1)$, $g(x) = \epsilon(x^2/\mu^2 - 1)$ with $x(0) = \sqrt{2}$, $\dot{x}(0) = 0$, $\omega = \lambda = \mu = 1$ and $T = 50$. The total energy $e = \dot{x}^2/2 + \omega^2(x^4/(4\lambda^2) - x^2)$ is not conserved.
The results presented in Fig. 10 correspond to $\epsilon = 0$ and indicate that (1) the full linearization method presented in this paper predicts positive displacements for $\Delta t = 0.1, 0.01, 0.001$ and $0.0001$, (2) the location of the first relative maximum of the displacement increases as the time step is decreased, (3) the locations of the relative maxima and minima of the velocity increase as the time step is decreased, (4) the phase diagram passes through the fixed point $(x, \dot{x}) = (0, 0)$ only for $\Delta t = 0.0001$ and $0.001$, whereas larger time steps do not preserve this fixed point, and (5) the energy is almost conserved for $\Delta t \leq 0.01$ but not for $\Delta t = 0.1$.

Eq. (21) was also solved with the MATLAB ode45 solver with tolerances equal to $10^{-5}$ and $10^{-13}$ [5]. For the higher tolerance, the ode45 solver predicted relative maximum displacements at about $t = 16.5$ and $t = 45$ and a negative relative minimum displacement at approximately $t = 31.5$, in marked contrast with the results presented in Fig. 10. However, for the smaller tolerance, it was found that the displacement was always greater than or equal to zero, dropped from its initial value to almost a nil value at about $t = 5$ and then exhibited a relative maximum at about $t = 34$. The displacement presented in Fig. 10 is

![Fig. 10. Example 7: displacement (top left), velocity (top right), phase diagram (bottom left) and nondimensional energy $E = e/e(0)$ (bottom right) as functions of time. $\omega = \lambda = \mu = 1$, $\epsilon = 0$, $x(0) = \sqrt{2}$, $\dot{x}(0) = 0$. Solid line: $\Delta t = 0.1$; dashed line: $\Delta t = 0.01$; dashed-dotted line: $\Delta t = 0.001$; dotted line: $\Delta t = 0.0001$.](image-url)
always greater than or equal to zero and is in accord with those of the MATLAB ode45 solver with a tolerance equal to $10^{-13}$, but the results of the full linearization method were found indistinguishable from those of the ode45 solver only for $\Delta t = 10^{-8}$.

Fig. 11 presents the results corresponding to the numerical solution of Eq. (21) with $\epsilon = 0, x(0) = \sqrt{2} + 0.1$ and $\dot{x}(0) = 0$, and indicates that there are very few differences between the results obtained with $\Delta t = 0.1, 0.01, 0.001$ and $0.0001$, although the larger time steps predict displacements and velocities whose phase increases as time is increased, and energies that do not remain constant. Note that the solution presented in Fig. 11 does not pass through the origin of the phase plane, and that the non-standard difference Euler method of Kojouharov and Welfert [5] predicts a phase acceleration and lower energy levels with $\Delta t = 0.1$ than those obtained with the MATLAB ode45 solver with a tolerance equal to $10^{-13}$.

The results presented in Fig. 12 correspond to Eq. (21) with $\epsilon = 0.2$ and indicate that (1) both the displacement and velocity are damped oscillations, (2) the numerical results for $\Delta t = 0.01$ are almost indistinguishable from those

Fig. 11. Example 7: displacement (top left), velocity (top right), phase diagram (bottom left) and nondimensional energy $E = e/e(0)$ (bottom right) as functions of time. $\omega = \lambda = \mu = 1$, $\epsilon = 0$, $x(0) = \sqrt{2} + 0.1$, $\dot{x}(0) = 0$. Solid line: $\Delta t = 0.1$; dashed line: $\Delta t = 0.01$; dashed-dotted line: $\Delta t = 0.001$; dotted line: $\Delta t = 0.0001$. 
corresponding to $\Delta t = 0.001$ an 0.0001, (3) the energy decreases as a function of time and its rate of decrease is almost independent of the time step for $\Delta t \leq 0.01$, (4) a time step equal to 0.1 results in phase acceleration of both the displacement and velocity, higher relative maxima and shallower relative minima and broader phase diagrams than $\Delta t = 0.001$, and (5) the numerical results obtained with the full linearization method and a time step equal to 0.0001 are indistinguishable from those obtained with the MATLAB variable-order ode45 solver with a tolerance of $10^{-13}$.

The results presented in Figs. 1–12 and, in particular, Figs. 9–12, indicate that the accuracy of full linearization methods which employ a fixed time step depends on the characteristics of the second-order odes to be solved, e.g., time-dependent or constant coefficients, nonlinearities, stiffness, conservative or dissipative properties, etc. These figures also show that the use of large time steps in time linearization methods depends on the characteristics of nonlinear second-order odes.

The results presented in previous paragraphs indicate that, for the same accuracy, the full linearization method presented in this paper may require
smaller (fixed) time steps than the MATLAB variable-order solvers, non-
standard Euler techniques and variable-order state transition matrix algo-
rithms for one-degree of freedom systems. This inconvenience is somewhat
reduced by the fact that the full linearization method provides explicit, albeit
complex nonlinear mappings for the discrete values of the displacement and
velocity. In addition, the method provides piecewise analytical solutions which
are globally differentiable. These two advantages are to be compared with both
the need for iterations of the MATLAB variable-order solvers and non-
standard Euler techniques and the the fact that these methods do provide discrete
solutions. Moreover, full linearization methods may use variable time steps
because their finite difference expressions depend on the linearization of the
nonlinear terms with respect to the displacement and velocity and time.

Despite the explicit, nonlinear finite difference form for the nodal values of
the displacement and velocity, full linearization methods provide, in exact
arithmetic, the exact solution of constant-coefficients odes with right-hand
sides that depend in a linear manner on the independent variable. In finite-
precision arithmetic, these methods are subject to round-off errors which may
be important when evaluating the exponential terms that appear in Eqs. (8)
and (9).

Full linearization methods are also stable. For the linear first-order ode
\[
\dot{u} = \lambda u,
\]
where \(\lambda\) is a complex constant, full linearization methods yield
\[
u_{n+1} = u_n \exp(\lambda \Delta t),
\]
and are (linearly) stable provided that \(\lambda_R \Delta t < 0\), i.e., they are \(A\)-stable, where
\(\lambda_R\) denotes the real part of \(\lambda\).

For the linear second-order ode
\[
\ddot{u} + D\dot{u} + Ku = 0,
\]
where the damping, \(D\), and stiffness, \(K\), are positive (real) constants, it is an
easy exercise to show that Eqs. (8) and (9) can be written as
\[
\mathbf{u}_{n+1} = \mathbf{M}\mathbf{u}_n,
\]
where \(\mathbf{u} = (u, \dot{u})^T\), the superscript \(T\) denotes transpose, and the matrix \(\mathbf{M}\) has
the following elements:
\[
M_{11} = \frac{\lambda^+ \exp(\lambda^- \Delta t) - \lambda^- \exp(\lambda^+ \Delta t)}{\lambda^+ - \lambda^-},
\]
\[
M_{12} = \frac{\exp(\lambda^+ \Delta t) - \exp(\lambda^- \Delta t)}{\lambda^+ - \lambda^-},
\]
\[
M_{21} = \lambda^+ \exp(\lambda^- \Delta t) - \exp(\lambda^+ \Delta t) \over \lambda^+ - \lambda^- ;
\]

\[
M_{22} = \lambda^+ \exp(\lambda^- \Delta t) - \lambda^- \exp(\lambda^+ \Delta t) \over \lambda^+ - \lambda^- ;
\]

where \(\lambda^\pm = -\frac{D}{2} \pm (\frac{D}{2})^2 - K \mid \). Therefore, a necessary and sufficient condition for linear stability is that the spectral radius of \(M\) be less than one, i.e., \(|\lambda_R| \Delta t < 0\) where \(|\lambda_R|\) denotes the largest real part of \(\lambda^\pm\). Therefore, full linearization methods are \(A\)-stable. If \(D = 0\), the elements of the matrix \(M\) are

\[
M_{11} = \cos(\lambda \Delta t), \quad M_{12} = \frac{1}{\lambda} \sin(\lambda \Delta t),
\]

\[
M_{21} = -\lambda \sin(\lambda \Delta t), \quad M_{22} = \cos(\lambda \Delta t),
\]

where \(\lambda^2 = K\), and its eigenvalues are complex conjugate and have modulus equal to one.

For systems of nonlinear second-order odes, i.e.,

\[
\ddot{u} = f(t, u, \dot{u}),
\]

where \(u\) and \(f\) are \(n\)th dimensional real vectors, full linearization techniques yield the following system of linear second-order odes:

\[
\ddot{u} + D \dot{u} + K u = h_n + g_n(t - t_n), \quad t_n < t \leq t_{n+1},
\]

where \(h\) and \(g\) are \(n\)th dimensional vectors and \(D\) and \(K\) are \(n \times n\) matrices, and since, in general, there is not a single matrix \(P_n\) that either diagonalizes or triangularizes (by means of Schur’s normal theorem) simultaneously both \(D\) and \(K\), Eq. (33) cannot be written, in general, as a system of \(n\) uncoupled linear second-order odes such as that of Eq. (3). Moreover, if Eq. (33) were written as a system of \(2n\) first-order odes, its solution would require the evaluation of a matrix exponential which is both time consuming and prone to numerical errors. This limits the applicability of full linearization methods to single-degree of freedom problems. However, systems of nonlinear second-order odes can still tackled with the partial linearization techniques presented in this paper. For example, if the linearization is performed with respect to only \(u\) and \(t\), then Eq. (33) reads

\[
\ddot{u} + K u = h_n + g_n(t - t_n), \quad t_n < t \leq t_{n+1},
\]

and one can always find an invertible matrix \(P_n\) such that \(P_n^{-1}K_nP_n\) is triangular (Schur’s normal form theorem or Jordan’s canonical form), so that (sequentially coupled) linear second-order odes result for all the components of \(v = P_n u\) and, therefore, explicit finite difference expressions can be obtained for the nodal values of \(v\). Unfortunately, this strategy requires the evaluation of \(P_n\) at each time step which is a time-consuming task if either Schur’s normal form theorem is used to determine the unitary matrix \(P_n\) or Jordan’s canonical forms.
are employed; in the latter, one is forced to determine the eigenvalues and eigenvectors of $K_n$. Similar comments apply if the linearization is performed with respect to only $u$ and $t$ which yields

$$u + D_n u = h_n + g_n (t - t_n), \quad t_n < t \leq t_{n+1}. \quad (35)$$

Alternatively, one can solve Eq. (33) iteratively by means of the procedure described in the previous paragraph, but the efficiency of the resulting method could be very low because, in addition to the determination of $P_n$, iterations would be required to achieve convergence. Thus, both the full and the partial linearization methods presented in this paper are mostly applicable to single-degree of freedom systems.

4. Conclusions

Non-standard finite difference methods for nonlinear second-order ordinary differential equations based on time linearization have been developed. These methods provide piecewise analytical solutions which are globally differentiable, and yield explicit, nonlinear mappings for the displacement and velocity at time level $n$ in terms of the same variables at the previous time level. These mappings depend in a nonlinear manner on the derivatives of the forcing, damping and elastic characteristics of the ode with respect to the displacement, velocity and time, and, therefore, may use a variable step size which may be adapted to the evolution of the solution.

In this paper, full linearization methods with fixed step size have been applied to seven nonlinear dynamical problems and it has been found that the accuracy of linearization methods degrades as the time step is increased. It has also been found that these methods provide results in good accord with those obtained with the MATLAB variable-order ode45 solver for the displacement, velocity, phase diagram and energy when they employ time steps on the same order as or smaller than the average step size used by the latter. However, they do require smaller time steps than variable-order state matrix algorithms which employ Neumann series and non-standard Euler methods.

It has also found that the use of large time steps in linearization methods depends on the characteristics of nonlinear second-order ordinary differential equations, i.e., conservative or dissipative, stiff or non-stiff, regular or chaotic equations, and are limited to single-degree of freedom problems.

Acknowledgements

This research was partially financed by Project BFM2001-1902 from the Ministerio de Ciencia y Tecnología of Spain and fondos FEDER.
References