Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 4 (2024)

A Semi-Analytical Solution of a Nonlinear Mass-Spring Finite Element Time-Dependent System

Professor Modify Andrew Elton Kaunda[1] and Professor Freddie Liswaniso Inambao[2]

[1]*Professor, Formerly of Cape Peninsula University of Technology, Bellville 7530, South Africa,*

[2]*Professor, University of KwaZulu-Natal, Durban 4001, South Africa.*

Abstract:

Second-order nonlinear mass-spring finite element time-dependent systems occurring in science and engineering are considered which generally do not have closed-form solutions and are solved using explicit incremental semi-analytical numerical solution procedures for nonlinear multiple-degree-of-freedom systems. Higher-order equivalent differential equations are derived to enable subsequent values of vectors to be updated using explicit Taylor series expansions. As the time step tends to zero, the values of displacement and velocity are exact in the Taylor series expansions involving as many higher-order derivatives as necessary. The ratio test is done for both the displacement and velocity Taylor series, to automatically adjust the size of time increments to ensure the algorithm's convergency, accuracy and stability. A linear system of two degrees of freedom was initially solved to illustrate how to extend the methods to deal with multiple degrees of freedom systems using matrices and vectors, typically obtained in finite element methods. The incremental semi-analytical solution procedures for nonlinear multiple-degree-of-freedom systems may be used to check results generated by implicit iterative procedures. Further applications of the semi-analytical procedures to time-dependent systems may be extended to time-independent systems that are differentiable in independent variables, such as partial differential equations with many independent variables.

Keywords:

Nonlinear dynamical systems; nonlinear oscillatory systems; higher-order equivalent differential equations; semi-analytical procedures.

1 Introduction

The study of natural sciences uses experimentation and observation to understand, describe, and predict thenatural world. It includes the study of many subjects, such as biology, chemistry, physics, astronomy, and earthscience. Integral and differential equations are formulated which are usually nonlinear. Such system equationsinclude the van der Pol equation, Lorenz equations, and Schoedinger equations in quantumphysics. The aims andobjectives of this study are to accurately solve nonlinear differential equations arising from the mathematicaltheories that describe and predict the natural world, where closed-form solutions are not possible, but linearisedequations and iterative solution procedures are usually employed.

This article gives details of robust semi-analytical numerical solution procedures for some nonlinear mass-spring finite element time-dependent systems occurring in science and engineering which generally do not have closed-form solutions. Higher-order equivalent differential equations are derived to enable subsequent values of vectors to be updated using explicit Taylor series expansions. As the time step tends to zero, the values of displacement and velocity are exact in the Taylor series expansions involving as many higher-order derivatives as necessary. The ratio test is done for both the displacement and velocity Taylor series, to automatically adjust the size of time increments to ensure the algorithm's convergency, accuracy and stability. A linear system of two degrees of freedom is initially solved to illustrate how to extend the methods to deal with multiple degrees of freedom systems using matrices and vectors, typically obtained in finite element methods. Starting by acknowledging the early remarkable contributions of the Newmark trapezoidal scheme [1], which possessed limited accuracy and stability characteristics, that was used for time-integration of nonlinear finite element analysis of solids and structures, to improve solution procedures, such as the improved numerical dissipation of Hilber et. al. [2], consistent tangent operators of Simo and Taylor [3], time-stepping schemes of Wood [4], simple second-order accurate implicit integration schemes of Bathe et. al. [5], and finite element methods of Zienkiewicz et. al. [6]. More contributions on convergence, stability and accuracy are cited in subsequent sections.

1.1 Implicit schemes

Zienkiewicz et. al. [6] introduced an implicit generalized Newmark integration scheme from the truncated Taylor series expansion of the displacement function u and its derivatives, as follows:

$$u_{n+1} = u_n + \Delta t \dot{u}_n + \cdots + \frac{\Delta t^p}{p!} u_n^{(p)} + \beta_p \frac{\Delta t^p}{p!}\left(u_{n+1}^{(p)} - u_n^{(p)}\right)$$

$$\dot{u}_{n+1} = \dot{u}_n + \Delta t \ddot{u}_n + \cdots + \frac{\Delta t^{p-1}}{(p-1)!} u_n^{(p)} + \beta_{p-1} \frac{\Delta t^{p-1}}{(p-1)!}\left(u_{n+1}^{(p)} - u_n^{(p)}\right)$$

(1)

$$u_{n+1}^{(p-1)} = u_n^{(p-1)} + \Delta t u_n^{(p)} + \beta_1 \Delta t \left(u_{n+1}^{(p)} - u_n^{(p)}\right)$$

where$u, \dot{u}, \ddot{u}$aredisplacement,velocityandacceleration.Setting$p = 2$formstheequivalentNewmarkscheme[1] which consists of two recurrence equations of displacement and velocity, and when combined with the governing second-order differential equation (4), gives three simultaneous equations in three unknowns. Carrying on from these contributions, a forward-backward difference time-integration scheme was developed by Kaunda [7], using the Taylor series, for solutions of nonlinear oscillatory systems, giving birth to more accurate implicit generalized one-step multiple-value algorithms [7],[8], repeated here for convenience.

$$s_{n+1} + \sum_{k=1}^{k=p}\left[\frac{(-1)^k}{k!}\left[\gamma_{1k}\Delta t \frac{d}{dt}\right]^k s_{n+1}\right] = s^* = s_n + \sum_{k=1}^{k=p}\left[\frac{1}{k!}\left[\beta_{1k}\Delta t \frac{d}{dt}\right]^k s_n\right]$$

(2)

$$v_{n+1} + \sum_{k=1}^{k=p-1}\left[\frac{(-1)^k}{k!}\left[\gamma_{2k}\Delta t \frac{d}{dt}\right]^k v_{n+1}\right] = v^* = v_n + \sum_{k=1}^{k=p-1}\left[\frac{1}{k!}\left[\beta_{2k}\Delta t \frac{d}{dt}\right]^k v_n\right]$$

(3)

where $s = x$ denotes displacement, $v = \dot{x}$ ˙ denotes velocity and $a = \ddot{x}$ ¨represents acceleration. Equations (2) and (3) provide the necessary extra equations to solve the differential equation (4) such that there are three equations in three unknowns. The implicit algorithms presented in [6],[7],[8], permitted to determine and optimize stability and accuracy of the recurrence equations by choosing appropriate tuneable integration parameters, $\beta_p, \gamma_{ik}, \beta_{ik}$. Numerical dissipation or algorithmic damping, mostly desired in finite element methods, may also be incorporated to filter out high-frequency responses, as considered in Hilber et. al. [2].

1.2 Explicit schemes

With supporting literature [6]-[18], on convergence, stability and accuracy, new semi-analytical procedures are now proposed for nonlinear multiple-degree-of-freedom systems with emphasis on reliable explicit incremental solution procedures, as opposed to iterative schemes, which turn out to be fast and accurate and depend on only differentiation (for continuously differentiable functions), as opposed to integration (usually difficult for nonlinear equations), to solve nonlinear differential equations. As a result, the stability of the algorithms is conditional and for small increments, convergence, stability and accuracy are simultaneously achieved. The explicit algorithm being focused in this paper is a subset of the implicit algorithms given by equations (1), (2) and (3), where $\beta_p =$

$0, \gamma_{ik} = 0, \beta_{ik} = 0$. Semi-analytical methods for the n-th order governing differential equations use higher-order equivalent differential equations. For example, the second-order differential equation (4), only displacement and velocity recurrence equations (2) and (3), which are associated with prescribed initial conditions, are updated using the Taylor series.

1.3 Article organization

The article is organized as follows: Section 2 develops the solution of nonlinear vector-valued oscillatory systems. Sections 3.1 and 3.2develop a two-degrees-of-freedom system, and extension to multiple-degree-of-freedom systems using mass, damping and stiffness matrices such as those obtained from finite element methods. Section 4 presents and discusses the results, and Section 5 draws conclusions.

2 Nonlinear vector-valued oscillatory system

The differential equation describing a nonlinear vector-valued oscillatory system may have the general form

$\ddot{x} + f(\dot{x}, x, t) = 0; x(0) = x_0; \dot{x}(0) = \dot{x}_0; t = t_0$(4)

The superposed dot on x, represents differentiation concerning time, t, and double-dot represents the second derivative. Closed-form solutions of most nonlinear systems do not exist. A semi-analytical solution of non-linear mass-spring finite element time-dependent systems occurring in science and engineering, which generally do not have closed-form solutions, is now considered.

2.1 Semi-analytical procedures for nonlinear vector-valued differential equations

For the above equations, in general homogeneous or non-homogeneous forms, the solution procedure is carried out as follows:

$$\ddot{x} = f(\dot{x}, x, t); x(0) = x_0; \dot{x}(0) = \dot{x}_0$$

$$\dddot{x} = \frac{d}{dt} f(\dot{x}, x, t) = \dot{f}(\ddot{x}, \dot{x}, x, t)$$

$$x^{(4)} = \frac{d^2}{dt^2} f(\dot{x}, x, t) = \ddot{f}(\dddot{x}, \ddot{x}, \dot{x}, x, t)$$

(5)

$$x^{(5)} = \frac{d^3}{dt^3} f(\dot{x}, x, t) = \dddot{f}\left(x^{(4)}, \dddot{x}, \ddot{x}, \dot{x}, x, t\right)$$

$$\cdots$$

$$x^{(N)} = \frac{d^{(N-2)}}{dt^{(N-2)}} f(\dot{x}, x, t)$$

These form higher-order equivalent differential equations [9] which are used in the solution of the waveform of the nonlinear vectors. Further higher-order derivatives may be necessary to increase accuracy, for $N \to \infty$. For the implicit iterative algorithms involving higher order derivatives exceeding the order of the differential equation, initial conditions of the vectors are determined at the beginning of each iteration, where the higher order equivalent differential equations come in handy. In contrast, for explicit algorithms, subsequent vector values of displacements, x_i, and velocities, $\dot{x}_{i,}$aredetermined and updated recursively from the explicit Taylor series expansions

$$x_{n+1} = x_n + \Delta t \dot{x}_n + \frac{\Delta t^2}{2!} \ddot{x}_n + \frac{\Delta t^3}{3!} \dddot{x}_n + \cdots;\ n = (0,1,2,3, \dots)$$

(6)

$$\dot{x}_{n+1} = \dot{x}_n + \Delta t \ddot{x}_n + \frac{\Delta t^2}{2!}\dddot{x}_n + \frac{\Delta t^3}{3!}x^{(4)} + \cdots;\; n = (0,1,2,3,\ldots)$$

where the analytically obtained acceleration and higher-order derivatives are evaluated from the higher-order equivalent differential equations (5). This is the first time in the procedure where errors are committed in the algorithm because of terminating the Taylor series at an upper summation limit of $N \ll \infty$.. Then, for each of the above equations, the solution procedure recursively proceeds as follows:

$$\ddot{x} = f(\dot{x}, x, t);\; x(n) = x_n; \dot{x}(n) = \dot{x}_n;\; n = (0,1,2,3,\ldots)$$

(7)

…

$$x^{(N)} = \frac{d^{(N-2)}}{dt^{(N-2)}} f(\dot{x}, x, t)$$

where the newly accepted sub-initial conditions are $x(n) = x_n; \dot{x}(n) = \dot{x}_n;\; n = (1,2,3,\ldots)$. Note that finitedifference methods, such as central difference, backward difference or implicit schemes, are not used in the above algorithm. The procedure is therefore an explicit incremental forward difference method using the Taylor series updates. Clearly, as the time-step, $\Delta t \to 0$, the values of displacement and velocity are exact in the Taylor series expansions involving as many higher-order derivatives as necessary, for $N \to \infty$. The explicit algorithm convergence, stability, accuracy and speed depend on the size of the time step and the number of higher-order derivatives included. For most practical examples, this is not a setback when compared with implicit algorithms. The convergence may be tested using the ratio test for both the displacement and velocity Taylor series which have to be updated. The stability is conditional depending on the size of the time step.

2.2 Practical termination of and error in recurrence equations

Ideally, the recurrence equations should be terminated at $N \to \infty$. Practically, for the most accurate results

$$x_{n+1} = x_n + \Delta t \dot{x}_n + \frac{\Delta t^2}{2!}\ddot{x}_n + \frac{\Delta t^3}{3!}\dddot{x}_n + \cdots + \frac{\Delta t^N}{N!}x_{\zeta_x}^{(N)};\; \zeta_x = n\left(1 + \frac{1}{N+1}\right) \in [n, n+1]$$

(8)

$$\dot{x}_{n+1} = \dot{x}_n + \Delta t \ddot{x}_n + \frac{\Delta t^2}{2!}\dddot{x}_n + \cdots + \frac{\Delta t^{N-1}}{(N-1)!}x_{\zeta_x}^{(N)};\; \zeta_{\dot{x}} = n\left(1 + \frac{1}{N}\right) \in [n, n+1]$$

where ζ_x represents the point where the last term is evaluated, at time, $t_\zeta$forthedisplacement, and similarly, $\zeta_{\dot{x}}$, for velocity, and $N$is the highest power of Δt for each series, respectively. The best estimate for ζ is elegantly derived in Irons et. al. [14]. The errors for displacement and velocity recurrence equations are, respectively, of order

$$\mathrm{O}\left(\frac{\Delta t^N}{N!}x_{\zeta_x}^{(N)}\right) \text{and } \mathrm{O}\left(\frac{\Delta t^{N-1}}{(N-1)!}x_{\zeta_{\dot{x}}}^{(N)}\right).$$

2.3 Radius of convergence and adaptive time stepping

The mathematical ratio test may be defined [18] for a power series cantered at $x = a$ by the radius of convergence

$$\lim_{n\to\infty} R = \frac{|C_n|}{|C_{n+1}|}$$

(9)

In this article, the following algorithm was adopted:

$$\lim_{n\to\infty} R_1 = \frac{|C_{n-1}|}{|C_n|}$$

(10)

$$\lim_{n\to\infty} R_2 = \frac{|C_{n-2}|}{|C_{n-1}|}$$

$$\lim_{n\to\infty} R_3 = \frac{|C_{n-3}|}{|C_{n-2}|}$$

Then, the radius of convergence adopted was, $R = min(R_1, R_2, R_3,, \dots)$.

1. If $R = \infty$, then the series converges for all x
2. If0 $< R < \infty$,thentheseriesconvergesforall$[x - a] < R$
3. If $R = 0$, then the series converges only for $x = a$

The ratio test needs to be done for both the displacement and velocity Taylor series, to automatically adjust the size of time increments to ensure convergency, accuracy and stability of the algorithm, for example, monitoring that the increment, $\Delta t < R$. If the time step is prescribed at the beginning of the algorithm such that $\Delta t \geq R$, the ratio test could be applied to adjust the time step appropriately, especially in the first-time increment. Repeating the ratio test after every time step may increase the overall execution time and cost of the algorithm. Some adaptive time stepping has been used by Y. Wang et. al. [19].

3 Vector-valued differential functionals

3.1 Two-degrees-of-freedom systems (2-dof)

A linear second-order

system subjected to a history of loading$f_i(t)$, initial conditions of displacement $x_i(0)$, and velocity $\dot{x}_i(0)$ forms a vector-valued function, and is shown in Figure 1.

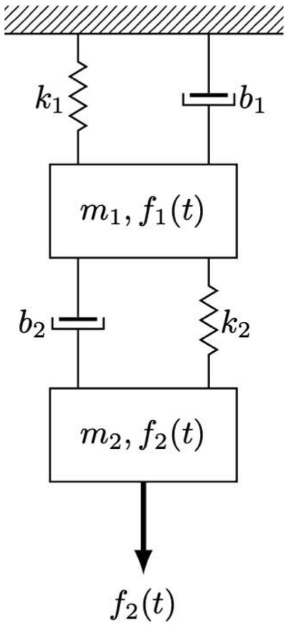

Figure 1: 2-dof: Mass-spring-damper system

Hence

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_{11} & k \\ k_{21} & k \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix}$$

(11)

which is differentiated once concerning time to get a third-order differential equation, and so on, to form higher-order equivalent differential equations [9].

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{Bmatrix} \dddot{x}_1 \\ \dddot{x}_2 \end{Bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_{11} & k \\ k_{21} & k \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{Bmatrix} \dot{f}_1 \\ \dot{f}_2 \end{Bmatrix}$$

(12)

After differentiating the third-order differential equation concerning time a fourth-order system is obtained, and further differentiation results in higher-order differential equations

$$\frac{d^N}{dt^N}\left\{\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_{11} & k \\ k_{21} & k \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix}\right\}$$

(13)

These form higher-order equivalent differential equations [9] which are used in the solution of the waveform of the nonlinear vectors, using the semi-analytical procedures considered in Section 2.1. Further higher-order derivatives may be necessary to increase accuracy. The eigenvalues λ, are determined from the eigenvalue problem or eigenproblem

$$Av = \lambda v$$

(14)

Where

$$A = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}^{-1} \begin{bmatrix} k_{11} & k \\ k_{21} & k \end{bmatrix}$$

(15)

and the corresponding natural frequencies are$\omega_i = \lambda_i^{\frac{1}{2}}$, and v is the eigenvector of the eigenproblem. Vector values of displacements, x_i, and velocities, $\dot{x}_i$ are then determined and updated recursively from the explicit Taylor series expansions

$$x_{n+1} = x_n + \Delta t \dot{x}_n + \frac{\Delta t^2}{2!}\ddot{x}_n + \frac{\Delta t^3}{3!}\dddot{x}_n + \cdots;\ n = (0,1,2,3,\dots)$$

(16)

$$\dot{x}_{n+1} = \dot{x}_n + \Delta t \ddot{x}_n + \frac{\Delta t^2}{2!}\dddot{x}_n + \frac{\Delta t^3}{3!}x^{(4)} + \cdots;\ n = (0,1,2,3,\dots)$$

where the exact vector values of acceleration and higher-order derivatives are evaluated from the higher-order equivalent differential equation.

3.2 Multiple-degrees-of-freedom systems (m-dof)

Vector-valued functions can be handled easily using matrices and vectors, such as mass matrices, M, damping matrices, B, and stiffness matrices, K, for example, obtained from finite-element methods such that

$$[M]\{a\} + [B]\{v\} + [K]\{s\} = \{f\} \qquad (17)$$

where $\{a\}, \{v\}, \{s\}, \{f\}$, are acceleration, velocity, displacement, and force vectors, respectively, and where the exact values of higher-order derivatives are evaluated analytically from the higher-order equivalent differential equations.

4 Discussion of results

Figure 2 shows a graph of displacement vs time for a linear two-degrees-of-freedom system taken from Thomson [16]with: $m_{11} = m_1 = 100, m_{12} = m_{21} = 0, m_{22} = m_2 = 25; b_{11} = b_{12} = b_{21} = b_{22} = 0; k_{11} = 54000, k_{12} = k_{21} = -18000$, $k_{22} = 18000$; $f_1 = 0, f_2 = 400$.

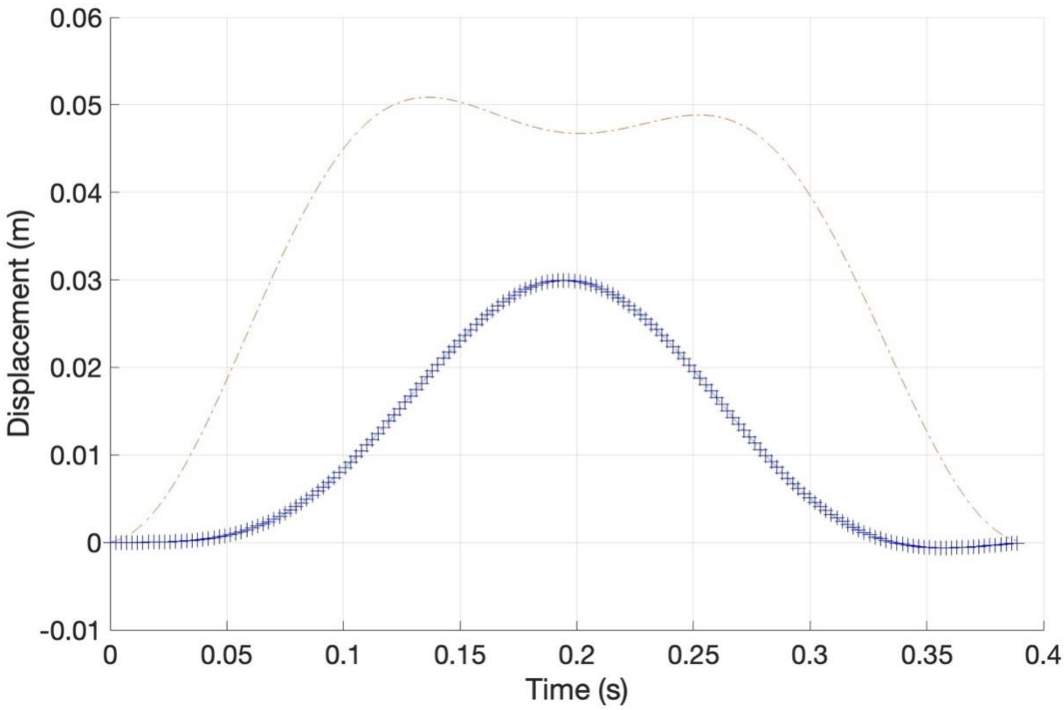

Figure 2: Graph of displacement vs time: 2-dof system

The eigenvalues were found as, $\lambda_1 = 258.9$, with the corresponding fundamental natural frequency, $\omega_1 = 16.09$ rad/s, and $\lambda_2 = 1001.1$ with the corresponding natural frequency, $\omega_2 = 31.6$ rad/s. The results given in Thomson [16] are confirmed.

A finite element example was taken from Y. Wang et. al. [19], which was a multiple-degree-of-freedom mass-spring system with up to 1500 nonlinear springs, whose details are shown in Table 1.

Table 1: Multiple-degree-of-freedom nonlinear mass-spring system

Mass (kg)	Spring (N/m)	Force (N)
$m_1 = 1$	$k_1 = k$	$f_1 = sin(t)$
m_2	k_2	$f_2 = sin(t)$
m_3	k_3	$f_3 = sin(t)$
...	...	...
m_n	k_n	$f_n = sin(t)$
	$k = 10^5$N/m	$\alpha = -2$

$$m_i = 1 \text{ kg}, \alpha = -2, k_i = k[1 + \alpha(u_i - u_{i-1})^2] \quad 2 \leq i \leq n$$

The system governing equations are given by those in Section 3.2, where each element of the spring matrix, K, is a nonlinear function of displacement, u_i, as shown in the table, and the system does not have damping.

Figure 3 shows a graph of displacement vs time for a 10-dof system solved using a fixed time step of $\Delta t = 1e - 03$(s). The CPU time was 2.197(s) for the duration of simulation of 10π(s) or 5 periodic cycles. The corresponding phase trajectory is shown in Figure 4, with a duration of 2π(s).

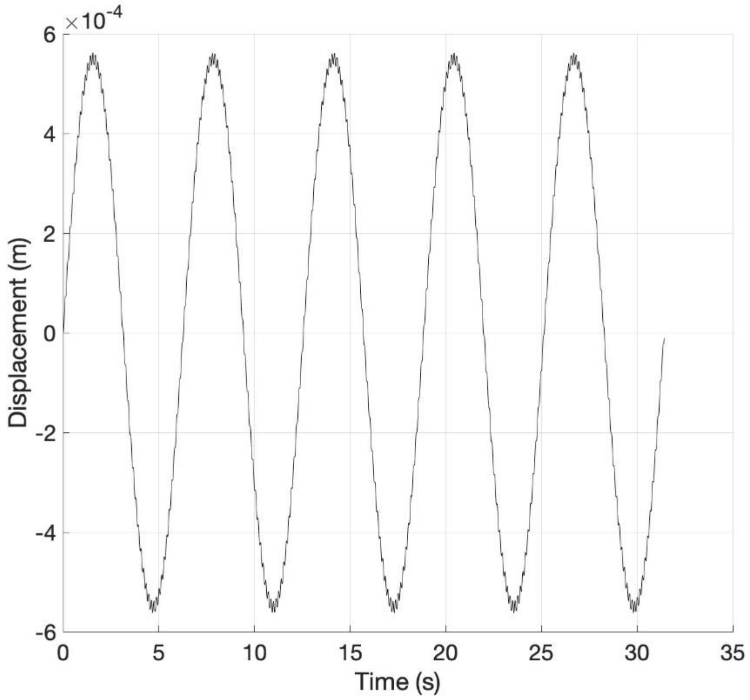

Figure 3: Graph of displacement vs time: 10-dof system

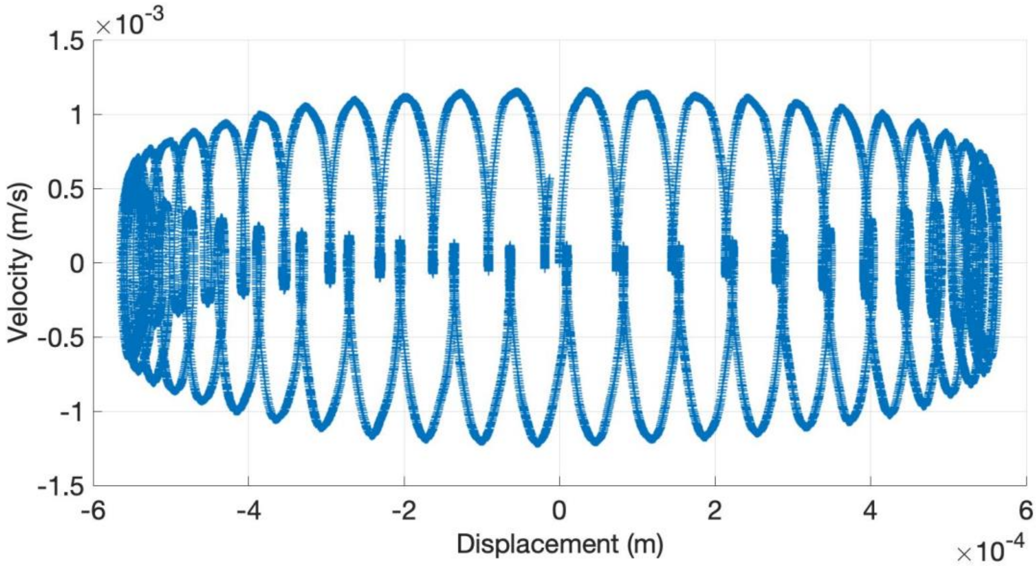

Figure 4: Graph of velocity vs displacement

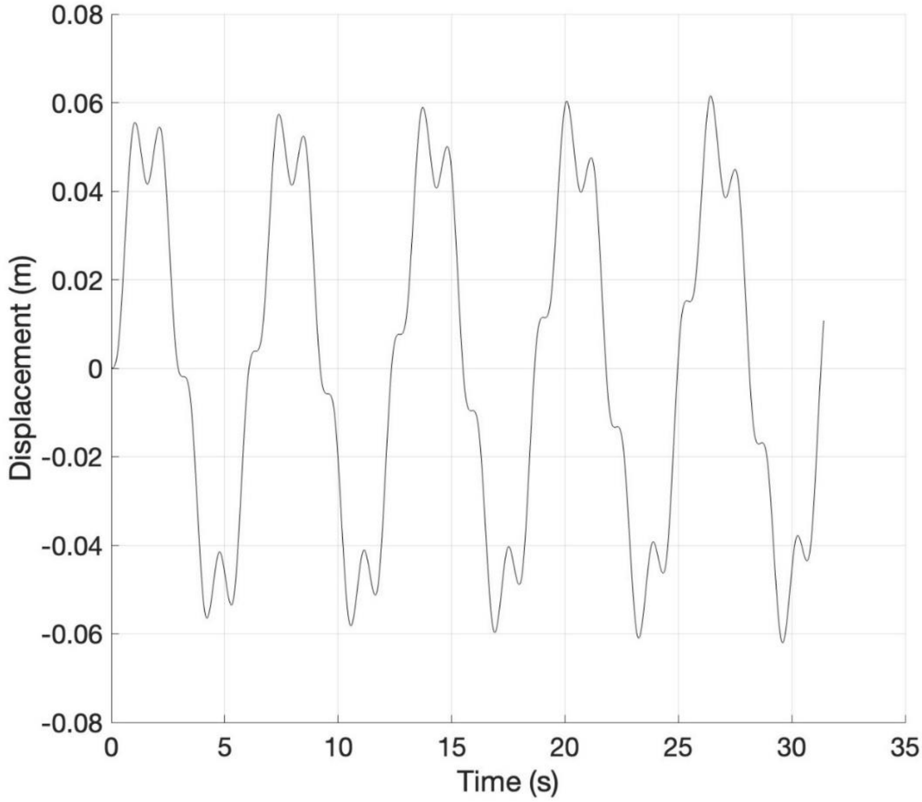

Figure 5: Graph of displacement vs time: 100-dof system

given a prescribed $\Delta t = 1e - 02$(s); with an adaptive time stepping scheme used was based on the radius of convergence and ratio test, $281.5945e - 06 \leq \Delta t \leq 1e - 03$(s). The CPU time was 254.616(s), whereas for a fixed $\Delta t = 1e - 3$(s), the CPU time was reduced to 69.094(s), for the same duration of simulation of 10π(s). The corresponding phase trajectory is shown in Figure 6, with a duration of 2π(s).

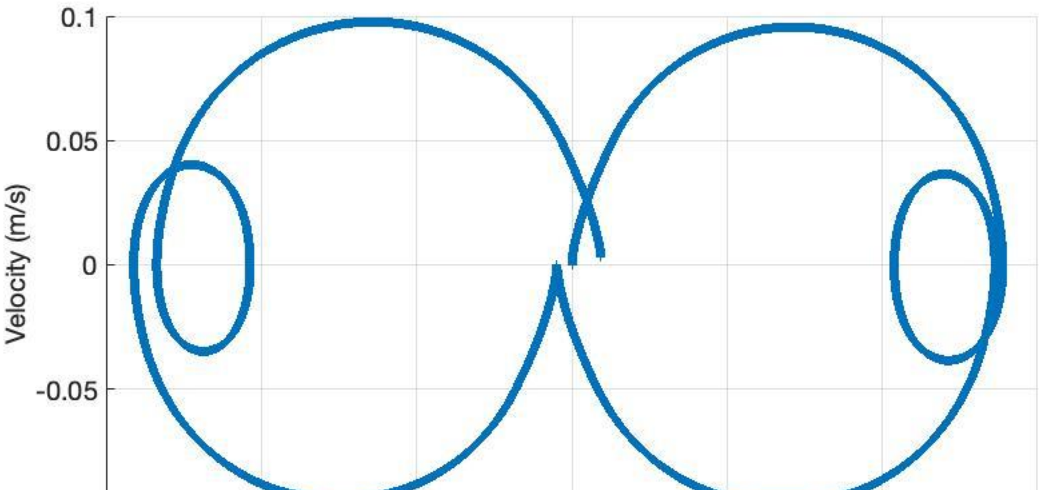

Figure 6: Graph of velocity vs displacement: 100-dof system

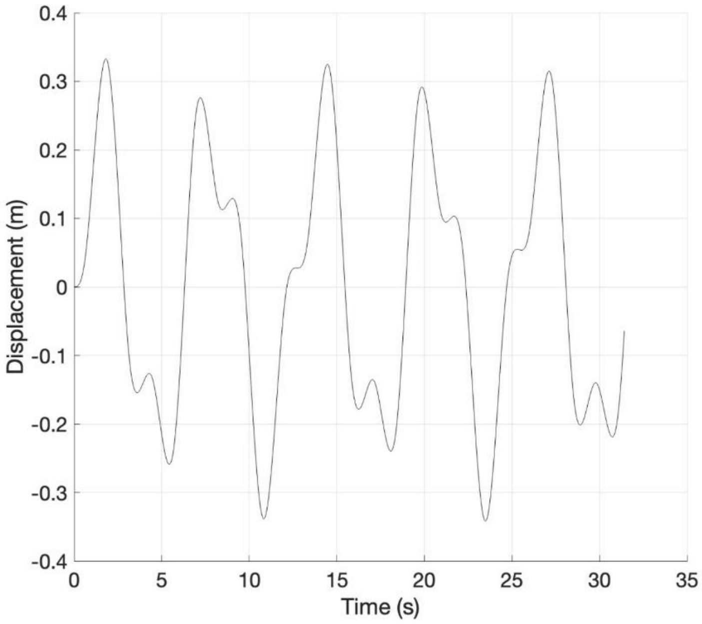

Figure 7: Graph of displacement vs time: 200-dof system

given a prescribed $\Delta t = 1e - 02$(s); with adaptive time stepping scheme used, 281.5945e − 06 ≤ Δt ≤ 1e − 03(s). The CPU time was 965.138(s), whereas a CPU time reduced to 258.764(s) for a fixed $\Delta t = 1e - 03$(s), for the same duration of simulation of 10π(s). The corresponding phase trajectory is shown in Figure 8, with a duration of 6π(s).

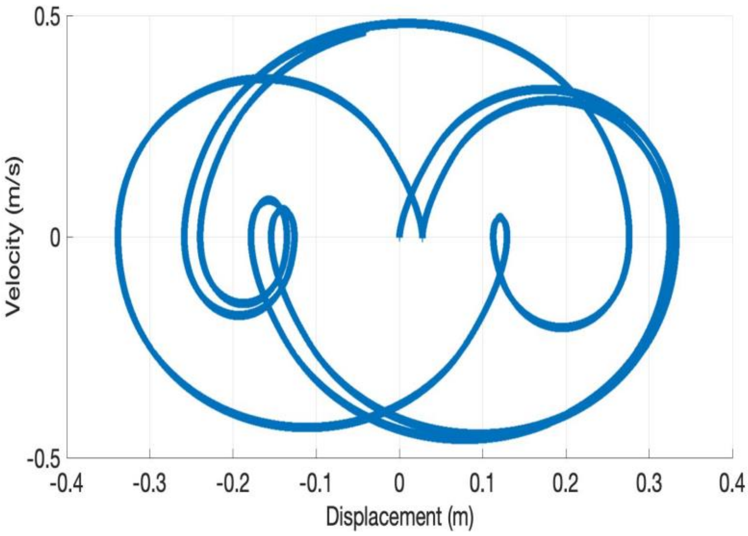

Figure 8: Graph of velocity vs displacement: 200-dof system

Figure 9 shows a graph of displacement vs time for: a 500-dof system; prescribed $\Delta t = 1e - 02$(s); with adaptive time stepping scheme used, 281.5945e − 06 ≤ Δt ≤ 1e − 03(s). The CPU time was, 5380.668(s), and for a fixed $\Delta t = 1e - 03$(s), the CPU time reduced to 1478.304(s), for the same duration of simulation of 10π(s). The graph shows that steady-state conditions were not reached, and the displacement kept on increasing in this duration.

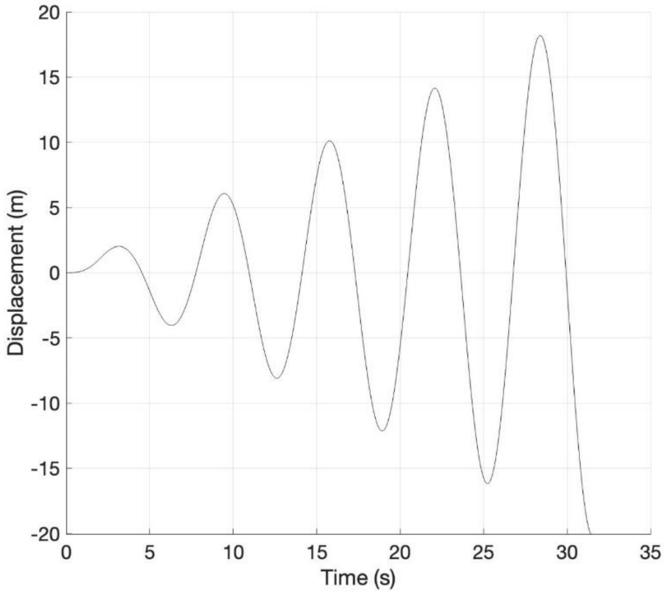

Figure 9: Graph of displacement vs time: 500-dof system

Figure 10 shows a graph of displacement vs time for a 1000-dof system solved using a fixed time step of $\Delta t = 1e-03$(s). The CPU time was 6485.748(s). The graph shows that steady-state conditions were reached within the duration of simulation of 10π(s).

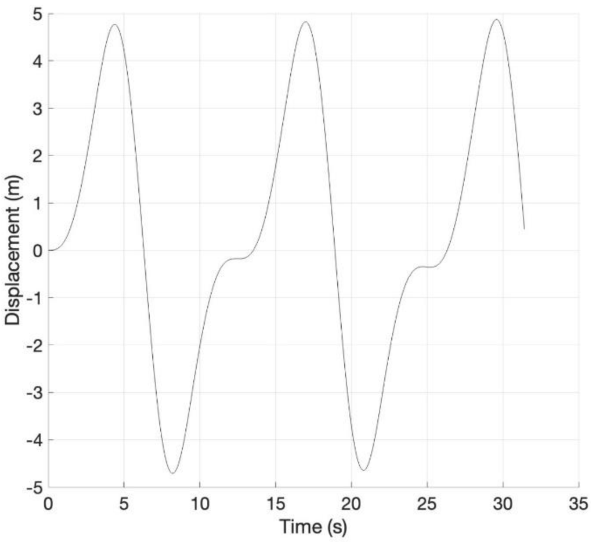

Figure 10: Graph of displacement vs time: 1000-dof system

While the CPU time is much greater than that of Y. Wang et. al. [19] having a time-step of 0.01(s), the graphs showed the same results, supporting the proof of concept for this incremental explicit method of solution which inevitably requires small time steps for both accuracy and stability. The theoretical truncation errors for displacement and velocity Taylor series recurrence equations used, were respectively, of $O\left(\frac{\Delta t^N}{N!} x_{\zeta_x}^{(N)}\right)$and $O\left(\frac{\Delta t^{N-1}}{(N-1)!} x_{\zeta_{\dot{x}}}^{(N)}\right)$, with $N = 10$. The graphs revealed that for a small number of degrees of freedom, such as 10-dof, the response is sinusoidal, whereas for many degrees of freedom, for example, greater than 100-dof, the responses departed from the sinusoidal curves. For the examples solved here, the fixed prescribed time step resulted in less CPU time as shown in Table 2, mostly because, for the adaptable time stepping implemented, far smaller time steps were applied automatically, based on the radius of convergence and ratio test for optimal accuracy and stability.

Table 2: CPU time for a duration of 10π seconds

Degree of freedom	10	100	200	500	1000
Δt (s)	$1e-3$	$1e-3$	$1e-3$	$1e-3$	$1e-3$
Total time (s)	2.197	69.094	258.764	1478.304	6485.748
min $\Delta t(s)$	$1e-3$	$281.5945e-6$	$281.5945e-6$	$281.5945e-6$	$1e-3$
max $\Delta t(s)$	$1e-3$	$1e-3$	$1e-3$	$1e-3$	$1e-3$
Total time (s)	2.197	254.616	965.138	5380.668	6485.748

5 Conclusions

Second order, $\ddot{x} = f(\dot{x}, x, t)$, nonlinear mass-spring finite element time-dependent systems have been solved using explicit incremental semi-analytical solutions for nonlinear multiple-degree-of-freedom systems. Higher order equivalent differential equations were formulated and then subsequent values of vectors were updated using explicit Taylor series expansions. Clearly, as the time-step, $\Delta t \to 0$, the values of displacement and velocity are exact in the Taylor series expansions involving as many higher order derivatives as necessary. The ratio test was done for both the displacement and velocity Taylor series, for the purpose of automatically adjusting the size of time increments to ensure convergency, accuracy and stability of the algorithm, for example, monitoring that the increment, $\Delta t < R$. If the time step was prescribed at the beginning of the algorithm such that $\Delta t \geq R$, the ratio test could be applied to adjust the time-step appropriately.

A linear system of two-degrees-of-freedom, taken from Thomson [16], was initially solved to illustrate how to extend the methods to deal with multiple-degrees-of-freedom systems using matrices and vectors, which are typically obtained in finite element methods, as shown in Table 1 with corresponding results shown in Figures 3, 5, 7, 9 and 10.

The incremental semi-analytical solution procedures for nonlinear multiple-degree-of-freedom systems may be used to check results generated by implicit iterative procedures. It is recommended that further applications of the semi-analytical procedures to time-dependent systems be extended to time-independent systems that are differentiable in terms of independent variables, such as partial differential equations having many independent variables.

References

[1] Newmark, Nathan M. (1959), ”A method of computation for structural dynamics”, Journal of Engineering Mechanics, ASCE, 85 (EM3): 67-94.

[2] Hilber, H. M., Hughes, T. J. R., and Taylor, R. L. ”Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics”, Earthquake Engineering and Structural Dynamics, 5 (1977), 283-292.

[3] Simo, J.C. and Taylor, R.L. ”Consistent tangent operators for rate-independent elastoplasticity”, Computer Methods in Applied Mechanics and Engineering, 48 (1985), 101-118, North-Holland.

[4] Wood, WL. Practical time-stepping schemes. Oxford, UK: Claredon Press; 1990.

[5] Bathe, K-J., Noh, G. ”Insight into an implicit time integration scheme for structural dynamics”. Computers and Structures, 98-99 (2012), 1-6.

[6] Zienkiewicz, O. C., Taylor, R. L. and Zhu, J. Z. The Finite Element Method: Its Basis and Fundamentals, 6th edition, Oxford, UK: Elsevier Butterworth-Heinemann; 2005.

[7] Kaunda, M. A. E. ”Forward-backward-difference Time-integrating Schemes with Higher Order Derivatives for Non-linear Finite Element Analysis of Solids and Structures”, Computers and Structures, 153, (2015), 1-18. http://doi.org/10.1016/j.compstruc.2015. 02.026.

[8] Kaunda M. A. E. ”Improved numerical solutions of nonlinear oscillatory systems”. Int J Numer Methods Eng. 2019;1-17. https://doi.org/10.1002/nme.6292.

[9] Smith, J. M. Mathematical Modelling and Digital Simulation for Engineers and Scientists, John Wiley and Sons Inc. 1977.

[10] Collatz, L. The numerical treatment of differential equations. 2nd printing of 3rd edition, Berlin, Germany: Springer Verlag/GMBH; 1966.

[11] Hildebrand, F. B. Introduction to numerical analysis, 2nd edition, Mineola, NY: Dover Publications, Inc.; 1974.

[12] Strogatz, Steven H. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. Reading, MA: Perseus Books Publishers; 1994.

[13] La Salle, J., and Lefschetz, S. 1961. Stability by Liapunov’s direct method with applications, RIAS, Inc. Baltimore, Maryland, Academic Press Inc.

[14] Irons, B.M. and Shrive, N.G. 1987. Numerical Methods in Engineering and Applied Science - Numbers are Fun, Ellis Horwood, Chichester.

[15] Meirovitch, L. Fundamentals of Vibrations, McGraw-Hill Higher Education, International Edition, 2001.

[16] Thomson, W.T. Theory of Vibrations with Applications, 2nd edition, George Allen and Unwin Ltd., 1983.

[17] Jordan, D. W. and Smith, P. Nonlinear ordinary differential equations: an introduction for scientists and engineers, 4th edition, New York, NY: Oxford University Press Inc.; 2007.

[18] Kreyszig, E. Advanced engineering mathematics, 10th edition, Wiley, 2011.

[19] Y. Wang, Tong Zhang, Xuelin Zhang, Shengwei Mei, Ningning Xie, Xiaodai Xue. ”On an accurate A-posteriori error estimator and adaptive time stepping for the implicit and explicit composite time integration algorithms”, Computers and Structures, 266, (2022). http://doi.org/10.1015/j.compstruc.2022.106789.