Computational Mechanics 22 (1998) 337–343 © Springer-Verlag 1998

An implicit multi-time step integration method for structural dynamics problems

P. Smolinski, Y.-S. Wu

Abstract A multi-time step integration algorithm is developed based on the trapezoidal rule time integration method for finite element equations of motion. This algorithm uses nodal groups to partition the mesh into subdomains that are updated with different time steps. A stability analysis of the method shows that the scheme retains the unconditionally stable behavior of the trapezoidal rule and conserves the same pseudo energy as the parent algorithm. Several numerical examples are used to verify the stability of the method and to investigate the accuracy of the scheme.

1 Introduction

Mixed-time integration and multi-time step integrations are two variations of standard time integration methods. With mixed-time integration (Hughes and Liu 1978; Liu and Belytschko 1982, Liu and Lin 1982, Liu 1983, Park 1980) different integration algorithms are used in different subdomains of the mesh. Multi-time step integration methods (Belytschko and Mullen 1977; Belytschko 1980; Neal and Belytschko; Smolinski, Sleith and Belytschko 1996), also referred to as subcycling, use different time steps to update spatial subdomains in a finite element mesh. Such methods are most often used with conditionally stable integration algorithms with the subdomain time step being determined by local stability criterion. In this way, as opposed to single time step integration, the properties of an element or a small group of elements do not dictate the time step for the entire mesh. Multi-time step integration is also useful for unconditionally stable time integration methods in that it can be advantageous to use different time steps in different spatial regions of the mesh for improved accuracy while avoiding the expense of updating the entire mesh with a small time step.

For structural dynamics problems, most of the work on multi-time step algorithms has focused on the explicit central difference method (Belytschko and Mullen 1977; Belytschko 1980; Neal and Belytschko 1989; Belytschko and Lu 1993), but despite the fact that these algorithms have existed for some time, little progress has been made in the stability analysis of these methods. The work that has been done on the stability of such methods has been based upon making simplifying assumptions or has consisted of numerical studies of model problems. It should be noted that stability is the critical issue with explicit integration and that some of these methods have been shown to have weak instabilities which become apparent only after a very long time period (Belytschko 1996). Only recently has an explicit multi-time step algorithm has been proposed (Smolinski, Sleith and Belytschko 1996) for which a rigorous stability analysis has been derived. The issue of stability is also important for implicit multi-time step algorithms since it may not be the case that the method retains the unconditionally stable behavior of the parent algorithm. The questions regarding stability may be one reason why such methods have yet to gain widespread use and as these questions are answered it is believed that they will find increased popularity.

In this paper an implicit multi-time step algorithm is developed for linear structural dynamics problems based on the trapezoidal rule integration method. This algorithm uses a nodal partition for the definition of mesh subdomains. Employing a similar procedure as that used in for a central difference subcycling algorithm (Smolinski, Sleith and Belytschko 1996), the eigenvalues and eigenvectors of the updates are found and expanding the solution in terms of the eigenvectors an energy-like quantity is evaluated. If the energy-like quantity remains bound over time then so does the solution. It is shown that the proposed algorithm retains the unconditionally stable behavior of the strandard trapezoidal rule and that the method conserves the same energy-like quantity as the single step algorithm. Several numerical examples are evaluated to verify the stability analysis and to examine the accuracy of the multi-time step algorithm.

2 Governing equation and integration algorithm

In this paper the equation of motion for linear, undamped structural dynamics will be considered which is

$$\mathbf{Ma} + \mathbf{Kd} = \mathbf{f} \tag{1}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix and $\mathbf{a}$, $\mathbf{d}$, and $\mathbf{f}$ are the vectors of nodal accelerations, displacements and external forces, respectively. It will be assumed that $\mathbf{M}$ is diagonal and positive definite and that $\mathbf{K}$ is

Communicated by W-K. Liu, 21 March 1998

P. Smolinski, Y.-S. Wu
Department of Mechanical Engineering,
University of Pittsburgh, Pittsburgh, PA 15261

Correspondence to: P. Smolinski

The helpful discussions with Ted Belytschko and the support of an Alcoa Foundation Fellowship are gratefully acknowledged.

symmetric and positive semi-definite. The initial value problem consists of finding $\mathbf{d}(t)$ for $t \in [0, T]$ satisfying (1) and subject to the initial conditions

$$\mathbf{d}(t=0) = \mathbf{d}^0 \tag{2}$$

$$\mathbf{v}(t=0) = \mathbf{v}^0 \tag{3}$$

where $\mathbf{v}$ is the nodal velocity vector and $\mathbf{d}^0$ and $\mathbf{v}^0$ are given initial data.

The trapezoidal rule time integration algorithm (Belytschko and Hughes 1983) is the most popular integration method for linear structural dynamics problems and is given by the equations

$$\mathbf{M}\mathbf{a}^{n+1} + \mathbf{K}\mathbf{d}^{n+1} = \mathbf{f}^{n+1} \tag{4}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{\Delta t}{2}\left(\mathbf{a}^n + \mathbf{a}^{n+1}\right) \tag{5}$$

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{4}\left(\mathbf{a}^n + \mathbf{a}^{n+1}\right) \tag{6}$$

where Δt is the time step and a superscript is used to denote the time, for example $\mathbf{d}^n = \mathbf{d}(\Delta n t)$. Multiplying (6) by $\mathbf{K}$ and noting from (4) that $\mathbf{Kd} = \mathbf{f} - \mathbf{Ma}$, $\mathbf{d}$ can be eliminated from (6) to give

$$\left(\mathbf{M} + \frac{\Delta t^2}{4}\mathbf{K}\right)\mathbf{a}^{n+1} = \left(\mathbf{M} - \frac{\Delta t^2}{4}\mathbf{K}\right)\mathbf{a}^n - \Delta t \mathbf{K}\mathbf{v}^n + \mathbf{f}^{n+1} - \mathbf{f}^n \tag{7}$$

where $\mathbf{a}^{n+1}$ is computed from this relation and $\mathbf{v}$ and $\mathbf{d}$ are updated using (5) and (6). From the above equation it can be seen that a system of linear equations must be solved to compute $\mathbf{a}^{n+1}$ and for this reason the trapezoidal rule is referred to as an implicit integration method.

The multi-time step trapezoidal algorithm is based on a nodal partition in which the mesh is partitioned into nodal groups which are updated with different time steps. Here, to simplify the presentation, the two group case will be considered, however, the method can be extended to a greater number of groups. For a two partition system the nodal groups will be labelled A and B are integrated with time steps, Δt and $r\Delta t$, respectively, where the time step ratio, r, is an integer such that $r > 1$. Vectors of nodal quantities can be partitioned according to the nodal groups in the form

$$\mathbf{d} = \begin{Bmatrix} \mathbf{d}_A \\ \mathbf{d}_B \end{Bmatrix} \tag{8}$$

where $\mathbf{d}_A$ is a vector of dimension N_A, $\mathbf{d}_B$ is a vector of dimension N_B and $N = N_A + N_B$ is the total number of nodes in the mesh. The vector partition is used to define a matrix partition as

$$\mathbf{Kd} = \begin{bmatrix} \mathbf{K}_A \\ \mathbf{K}_B \end{bmatrix} \begin{Bmatrix} \mathbf{d}_A \\ \mathbf{d}_B \end{Bmatrix} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix} \begin{Bmatrix} \mathbf{d}_A \\ \mathbf{d}_B \end{Bmatrix} \tag{9}$$

or

$$\mathbf{Ma} = \begin{bmatrix} \mathbf{M}_A & 0 \\ 0 & \mathbf{M}_B \end{bmatrix} \begin{Bmatrix} \mathbf{a}_A \\ \mathbf{a}_B \end{Bmatrix} . \tag{10}$$

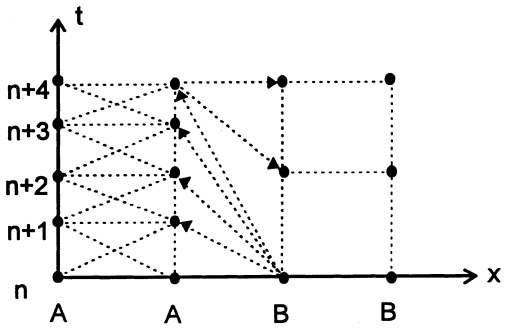

Fig. 1. Flow of information for the case $r = 2$

With this notation double subscripts are used to denote particular rows and columns of a matrix while a single subscript refers to particular rows of a matrix.

In this algorithm the two nodal groups are updated separately; first group A is updated $2r$ times with the time step Δt and then group B is updated twice with the time step $r\Delta t$. This set of updates advances the solution from step n to step $n + 2r$ in both nodal subdomains and is referred to as a total cycle. The flow of information during the total cycle for $r = 2$ is shown in Fig. 1 and the total cycle is repeated until the desired time history has been computed. Equations for the update of each nodal group will be developed in the following.

Nodal group *A* update

The equations for the group A update are derived by assuming a constant velocity in nodal group B, $\mathbf{v}_B^{n+1} = \mathbf{v}_B^n$, which from (5) can be achieved if $\mathbf{a}_B^{n+1} = -\mathbf{a}_B^n$. The change in $\mathbf{v}_B$ and $\mathbf{a}_B$ is termed a fictitious update in that these altered values are only used for the update of group A and the actual updated values will be calculated in the group B update. Given these approximations (5) and (7) can be reduced to

$$\left(\mathbf{M}_A + \frac{\Delta t^2}{4}\mathbf{K}_{AA}\right)\mathbf{a}_A^{n+1} = \left(\mathbf{M}_A - \frac{\Delta t^2}{4}\mathbf{K}_{AA}\right)\mathbf{a}_A^n - \Delta t \mathbf{K}_A \mathbf{v}^n + \mathbf{f}_A^{n+1} - \mathbf{f}_A^n \tag{11}$$

$$\mathbf{v}_A^{n+1} = \mathbf{v}_A^n + \frac{\Delta t}{2}\left(\mathbf{a}_A^n + \mathbf{a}_A^{n+1}\right) \tag{12}$$

and the displacement in group B is kept fixed so that (6) is rewritten as

$$\mathbf{d}_A^{n+1} = \mathbf{d}_A^n + \Delta t \mathbf{v}_A^n + \frac{\Delta t^2}{4}\left(\mathbf{a}_A^n + \mathbf{a}_A^{n+1}\right) . \tag{13}$$

After group A is updated $2r$ times using (11–13), the concomitant fictitious updates of group B results in $\mathbf{d}_B^{n+2r} = \mathbf{d}_B^n$, $\mathbf{v}_B^{n+2r} = \mathbf{v}_B^n$ and $\mathbf{a}_B^{n+2r} = \mathbf{a}_B^n$ so that in fact the nodal variables are unchanged.

Nodal group *B* update

The update of nodal group B is derived in a similar fashion to that of group A and by changing the subscript to B and replacing Δt by $r\Delta t$ in (11–13) the update equations are

$$\left(\mathbf{M}_B + \frac{(r\Delta t)^2}{4}\mathbf{K}_{BB}\right)\mathbf{a}_B^{n+r} = \left(\mathbf{M}_B - \frac{(r\Delta t)^2}{4}\mathbf{K}_{BB}\right)\mathbf{a}_B^n - r\Delta t\mathbf{K}_B\mathbf{v}^n + \mathbf{f}_B^{n+r} - \mathbf{f}_B^n \tag{14}$$

$$\mathbf{v}_B^{n+r} = \mathbf{v}_B^n + \frac{r\Delta t}{2}\left(\mathbf{a}_B^n + \mathbf{a}_B^{n+r}\right) \tag{15}$$

$$\mathbf{d}_B^{n+r} = \mathbf{d}_B^n + r\Delta t\mathbf{v}_B^n + \frac{(r\Delta t)^2}{4}\left(\mathbf{a}_B^n + \mathbf{a}_B^{n+r}\right) \tag{16}$$

and for the corresponding ficticious updates of group A: $\mathbf{d}_A^{n+r} = \mathbf{d}_A^n$, $\mathbf{v}_A^{n+r} = \mathbf{v}_A^n$ and $\mathbf{a}_A^{n+r} = -\mathbf{a}_A^n$. This update is performed twice in succession and after the ficticious updates the nodal quantities of group A are unchanged.

3 Stability analysis

To analyze the stability of a time integration algorithm it is sufficient to consider the homogeneous form of the equation of motion ($f = 0$). In which case the update equations of **a** and **v** for the single time step algorithm can be written in matrix form from (5) and (7) as

$$\begin{bmatrix} \mathbf{I} & -\frac{\Delta t}{2}\mathbf{I} \\ 0 & \mathbf{M} + \frac{\Delta t^2}{4}\mathbf{K} \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^{n+1} = \begin{bmatrix} \mathbf{I} & \frac{\Delta t}{2}\mathbf{I} \\ -\Delta t\mathbf{K} & \mathbf{M} - \frac{\Delta t^2}{4}\mathbf{K} \end{bmatrix} \begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^n \tag{17}$$

or

$$\mathbf{B}\mathbf{u}^{n+1} = \mathbf{A}\mathbf{u}^n \tag{18}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{I} & -\frac{\Delta t}{2}\mathbf{I} \\ 0 & \mathbf{M} + \frac{\Delta t^2}{4}\mathbf{K} \end{bmatrix} \tag{19}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \frac{\Delta t}{2}\mathbf{I} \\ -\Delta t\mathbf{K} & \mathbf{M} - \frac{\Delta t^2}{4}\mathbf{K} \end{bmatrix} \tag{20}$$

and

$$\mathbf{u} = \begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix} . \tag{21}$$

The generalized eigenvalue problem associated with (18) is given by

$$\mathbf{A}\mathbf{x} = \lambda\,\mathbf{B}\mathbf{x} \tag{22}$$

where λ is the eigenvalue and **x** is the eigenvector. Substituting the matrices **A** and **B** given by (19) and (20) into (22), it can be shown that there is a set of $2N$ eigenvectors of the form

$$\mathbf{x}_j = \begin{Bmatrix} \frac{\mathrm{i}}{\omega_j}\mathbf{z}_j \\ \mathbf{z}_j \end{Bmatrix},\ j = 1, N \tag{23}$$

$$\bar{\mathbf{x}}_j = \begin{Bmatrix} \frac{-\mathrm{i}}{\omega_j}\mathbf{z}_j \\ \mathbf{z}_j \end{Bmatrix},\ j = 1, N \tag{24}$$

where the overbar denotes a complex conjugate, i is the imaginary number and $\mathbf{z}_j$ and ω_j are given by the generalized eigenvalue problem

$$\mathbf{K}\mathbf{z}_j = \omega_j^2\mathbf{M}\mathbf{z}_j\ . \tag{25}$$

Since the matrices **K** and **M** are symmetric the eigenvectors $\mathbf{z}_j$ have the orthogonality properties

$$\mathbf{z}_j^{\mathrm{T}}\mathbf{M}\mathbf{z}_k = \delta_{jk} \tag{26}$$

$$\mathbf{z}_j^{\mathrm{T}}\mathbf{K}\mathbf{z}_k = \omega_j^2\delta a_{jk} \quad (\text{no sum on } j) \tag{27}$$

for $j, k = 1, N$ and span R^N. The complex conjugate eigenvalues, λ_j and $\bar{\lambda}_j$, can be found from the relation

$$\omega_j^2 = \frac{-4(\lambda_j - 1)^2}{\Delta t^2(\lambda_j + 1)^2} \tag{28}$$

and it can be shown that regardless of the time step

$$\lambda_j\bar{\lambda}_j = 1 \tag{29}$$

in which case λ_j can be taken as

$$\lambda_j = \cos\theta_j + \mathrm{i}\sin\theta_j\ . \tag{30}$$

Since the eigenvectors $\mathbf{x}_j$ and $\mathbf{x}_j$ span R^{2n}, $\mathbf{v}^n$ and $\mathbf{a}^n$ can be expanded in the form

$$\begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^n = \alpha_j\mathbf{x}_j + \bar{\alpha}_j\bar{\mathbf{x}}_j \tag{31}$$

and substituting the above equation into (18) and using (22) gives the result

$$\begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^{n+1} = \alpha_j\lambda_j\mathbf{x}_j + \bar{\alpha}_j\bar{\lambda}_j\bar{\mathbf{x}}_j\ . \tag{32}$$

Writing the complex coefficient α_j as

$$\alpha_j = A_j(\cos\phi_j + \mathrm{i}\sin\phi_j) \tag{33}$$

and substituting the above into (31) and (32) and simplifying gives the results

$$\mathbf{a}^n = 2A_j\cos\phi_j z_j \tag{34}$$

$$\mathbf{v}^n = 2\frac{A_j}{\omega_j}\sin\phi_j z_j \tag{35}$$

$$\mathbf{a}^{n+1} = 2A_j\cos(\phi_j + \theta_j)z_j \tag{36}$$

$$\mathbf{v}^{n+1} = 2\frac{A_j}{\omega_j}\sin(\phi_j + \theta_j)z_j\ . \tag{37}$$

To examine the stability of the integration algorithm a pseudo energy of the form

$$E = \mathbf{a}^{\mathrm{T}}\mathbf{M}\mathbf{a} + \mathbf{v}^{\mathrm{T}}\mathbf{K}\mathbf{v} \tag{38}$$

will be considered. It has been shown (Belytschko and Hughes 1983) that this quantity does not grow over a time step, i.e. $E^{n+1} = E^n$, which means that it invariant for all time. Given the properties of **M** and **K**, this implies that **a** and **v** remain bounded over time and hence the algorithm is stable. To show this, E is calculated at time steps n and

$n+1$ by substituting (34–37) into (38) and with the use of the orthogonality properties given in (26, 27) yields

$$E^n = E^{n+1} = \sum_{j=1}^{N} 4A_j^2 \tag{39}$$

and thus the pseudo energy remains unchanged after the update. Since this is true for any Δt, the algorithm is unconditionally stable.

The stability analysis of the multi-time step algorithm will be done in a similar fashion, however, it is necessary to show that this psuedo energy is conserved for both nodal group updates. This will be examined separately for the updates of the individual nodal groups as follows.

Group *A* update

The matrices **A** and **B** in (18) for the group *A* update will be denoted $\mathbf{A}_A$ and $\mathbf{B}_A$ and from (11) and (12) can be shown to be

$$\mathbf{B}_A = \begin{bmatrix} \mathbf{I}_A & 0 & -\frac{\Delta t}{2}\mathbf{I}_A & 0 \\ 0 & \mathbf{I}_B & 0 & 0 \\ 0 & 0 & \mathbf{M}_A + \frac{\Delta t^2}{4}\mathbf{K}_{AA} & 0 \\ 0 & 0 & 0 & \mathbf{I}_B \end{bmatrix} \tag{40}$$

$$\mathbf{A}_A = \begin{bmatrix} \mathbf{I}_A & 0 & \frac{\Delta t}{2}\mathbf{I}_A & 0 \\ 0 & \mathbf{I}_B & 0 & 0 \\ -\Delta t\mathbf{K}_{AA} & -\Delta t\mathbf{K}_{AB} & \mathbf{M}_A - \frac{\Delta t^2}{4}\mathbf{K}_{AA} & 0 \\ 0 & 0 & 0 & -\mathbf{I}_B \end{bmatrix}. \tag{41}$$

Substituting the above matrices into (22) and simplifying gives the eigenvalues and eigenvectors

$$\mathbf{x}_j = \begin{Bmatrix} \frac{\mathrm{i}}{\omega_j}\mathbf{z}_{Aj} \\ 0 \\ \mathbf{z}_{Aj} \\ 0 \end{Bmatrix}, \lambda_{Aj} \quad j = 1, N_A \tag{42}$$

$$\bar{\mathbf{x}}_j = \begin{Bmatrix} \frac{-\mathrm{i}}{\omega_{Aj}}\mathbf{z}_{Aj} \\ 0 \\ \mathbf{z}_{Aj} \\ 0 \end{Bmatrix}, \bar{\lambda}_{Aj} \quad j = 1, N_A \tag{43}$$

$$\mathbf{x}'_I = \begin{Bmatrix} -\mathbf{K}_{AA}^{-1}\mathbf{K}_{AB}\mathbf{z}_{BI} \\ \mathbf{z}_{BI} \\ 0 \\ 0 \end{Bmatrix}, \lambda_I = 1 \quad I = 1, N_B \tag{44}$$

$$\mathbf{x}''_I = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{z}_{BI} \end{Bmatrix}, \lambda_I = -1 \quad I = 1, N_B \tag{45}$$

where $\lambda_{Aj}, \mathbf{z}_{Aj}$ and ω_{Aj} are given by

$$\omega_{Aj}^2 = \frac{-4(\lambda_{Aj}-1)^2}{\Delta t^2(\lambda_{Aj}+1)^2} \tag{46}$$

$$\mathbf{K}_{AA}\mathbf{z}_{Aj} = \omega_{Aj}^2\mathbf{M}_A\mathbf{z}_{Aj} \tag{47}$$

and the vectors $\mathbf{z}_{BI}$ are arbitrary but assumed to span R^{N_B}. The matrices $\mathbf{K}_{AA}$ and $\mathbf{M}_A$ are symmetric and the eigenvectors $\mathbf{z}_{Aj}$ have the orthogonality properties

$$\mathbf{z}_{Aj}^{\mathrm{T}}\mathbf{M}_A\mathbf{z}_{Ak} = \delta_{jk} \tag{48}$$

$$\mathbf{z}_{Aj}^{\mathrm{T}}\mathbf{K}_{AA}\mathbf{z}_{Ak} = \omega_{Aj}^2\delta_{jk} \quad (\text{ no sum on j}) \tag{49}$$

for $j, k = 1, N_A$ and span R^{N_A}. Since from (46) it can be shown that $\lambda_{Aj}\bar{\lambda}_{Aj} = 1$, then λ_{Aj} can be written as

$$\lambda_{Aj} = \cos\theta_{Aj} + \mathrm{i}\sin\theta_{Aj} \ . \tag{50}$$

If the velocities and accelerations are written in terms of the eigenvectors (42–45) as

$$\begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^n = \alpha_j\mathbf{x}_j + \bar{\alpha}_j\bar{\mathbf{x}}_j + \beta_I\mathbf{x}'_I + \gamma_I\mathbf{x}''_I \tag{51}$$

then from (18) and (22) the updated values are given by

$$\begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^{n+1} = \alpha_j\lambda_j\mathbf{x}_j + \bar{\alpha}_j\bar{\lambda}_j\bar{\mathbf{x}}_j + \beta_I\mathbf{x}'_I + \gamma_I\mathbf{x}''_I \quad \text{sum on j, I} \tag{52}$$

and substituting **v** and **a** given in (51) and (52) into (38) and using the orthogonality conditions (48, 49) yields

$$E^n = \left(\alpha_j + \bar{\alpha}_j\right)^2 + \gamma_I\gamma_p\mathbf{z}_{BI}^{\mathrm{T}}\mathbf{M}_B\mathbf{z}_{Bp} + \left(\frac{\mathrm{i}\alpha_j}{\omega_{Aj}} - \frac{\mathrm{i}\bar{\alpha}_j}{\omega_{Aj}}\right)^2 + \beta_I\beta_p\mathbf{z}_{BI}^{\mathrm{T}}(\mathbf{K}'_{BB})\mathbf{z}_{Bp} \tag{53}$$

$$E^{n+1} = \left(\alpha_j\lambda_j + \bar{\alpha}_j\bar{\lambda}_j\right)^2 + \gamma_I\gamma_p\mathbf{z}_{BI}^{\mathrm{T}}\mathbf{M}_B\mathbf{z}_{Bp} + \left(\frac{\mathrm{i}\alpha_j\lambda_j}{\omega_{Aj}} - \frac{\mathrm{i}\bar{\alpha}_j\bar{\lambda}_j}{\omega_{Aj}}\right)^2 + \beta_I\beta_p\mathbf{z}_{BI}^{\mathrm{T}}(\mathbf{K}'_{BB})\mathbf{z}_{Bp} \tag{54}$$

where

$$\mathbf{K}'_{BB} = \mathbf{K}_{BB} - \mathbf{K}_{BA}\mathbf{K}_{AA}^{-1}\mathbf{K}_{AB}. \tag{55}$$

The expressions in (53) and (54) can be further reduced by using (50) and taking α_j as given in (33) to

$$E^n = 4A_j^2 + \gamma_I\gamma_p\mathbf{z}_{BI}^{\mathrm{T}}\mathbf{M}_B\mathbf{z}_{Bp} + \beta_I\beta_p\mathbf{z}_{BI}^{\mathrm{T}}(\mathbf{K}'_{BB})\mathbf{z}_{Bp} \tag{56}$$

$$E^{n+1} = 4A_j^2 + \gamma_I\gamma_p\mathbf{z}_{BI}^{\mathrm{T}}\mathbf{M}_B\mathbf{z}_{Bp} + \beta_I\beta_p\mathbf{z}_{BI}^{\mathrm{T}}(\mathbf{K}'_{BB})\mathbf{z}_{Bp} \tag{57}$$

and comparing the above two expression it is seen that $E^{n+1} = E^n$ for the group *A* update.

Group *B* update

The update matrices for the group *B* are given by (11) and (19) as

$$\mathbf{B}_B = \begin{bmatrix} \mathbf{I}_A & 0 & 0 & 0 \\ 0 & \mathbf{I}_B & 0 & -\frac{r\Delta t}{2}\mathbf{I}_A \\ 0 & 0 & \mathbf{I}_A & 0 \\ 0 & 0 & 0 & \mathbf{M}_B + \frac{(r\Delta t)^2}{4}\mathbf{K}_{BB} \end{bmatrix} \tag{58}$$

$$\mathbf{A}_B = \begin{bmatrix} \mathbf{I}_A & 0 & 0 & 0 \\ 0 & \mathbf{I}_B & 0 & \frac{r\Delta t}{2}\mathbf{I}_A \\ 0 & 0 & -\mathbf{I}_A & 0 \\ -r\Delta t\mathbf{K}_{AA} & -r\Delta t\mathbf{K}_AB & 0 & \mathbf{M}_B - \frac{(r\Delta t)^2}{4}\mathbf{K}_{BB} \end{bmatrix} \tag{59}$$

and substituting the above matrices into (22) and with some algebra gives

$$\mathbf{x}_j = \begin{Bmatrix} 0 \\ \frac{\mathrm{i}}{\omega_{Bj}}\mathbf{z}_{Bj} \\ 0 \\ \mathbf{z}_{Bj} \end{Bmatrix}, \lambda_{Bj} \quad j = 1, N_B \tag{60}$$

$$\bar{\mathbf{x}}_j = \begin{Bmatrix} 0 \\ -\frac{\mathrm{i}}{\omega_{Bj}}\mathbf{z}_{Bj} \\ 0 \\ \mathbf{z}_{Bj} \end{Bmatrix}, \bar{\lambda}_{Bj} \quad j = 1, N_B \tag{61}$$

$$\mathbf{x}'_I = \begin{Bmatrix} \mathbf{z}_{AI} \\ -\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA}\mathbf{z}_{AI} \\ 0 \\ 0 \end{Bmatrix}, \lambda_I = 1 \quad I = 1, N_A \tag{62}$$

$$\mathbf{x}''_I = \begin{Bmatrix} 0 \\ 0 \\ \mathbf{z}_{AI} \\ 0 \end{Bmatrix}, \lambda_I = -1 \quad I = 1, N_A \ . \tag{63}$$

In the above expressions the vectors $\mathbf{z}_{AI}$ are arbitrary and $\lambda_{Bj}, \mathbf{z}_{Bj}$ and ω_{Bj} are given by

$$\omega_{Bj}^2 = \frac{-4(\lambda_{Bj} - 1)^2}{r^2 \Delta t^2 (\lambda_{Bj} + 1)^2} \tag{64}$$

$$\mathbf{K}_{BB}\mathbf{z}_{Bj} = \omega_{Bj}^2 \mathbf{M}_B \mathbf{z}_{Bj} \tag{65}$$

where the eigenvectors $\mathbf{z}_{BI}$ are orthogonal with respect to the matrices $\mathbf{K}_{BB}$ and $\mathbf{M}_B$ such that

$$\mathbf{z}_{Bj}^{\mathrm{T}}\mathbf{M}_B\mathbf{z}_{Bk} = \delta_{jk} \tag{66}$$

$$\mathbf{z}_{Bj}^{\mathrm{T}}\mathbf{K}_{BB}\mathbf{z}_{Bk} = \omega_{Bj}^2\delta_{jk} \quad \text{(no sum on j)} \ . \tag{67}$$

The nodal velocities and accelerations at time step n are written in terms of the vectors given in (60–63) as

$$\begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^n = \alpha_j \mathbf{x}_j + \bar{\alpha}_j \bar{\mathbf{x}}_j + \beta_I \mathbf{x}'_I + \gamma_I \mathbf{x}''_I \tag{68}$$

and from (18) and (22) the updated values are given by

$$\begin{Bmatrix} \mathbf{v} \\ \mathbf{a} \end{Bmatrix}^{n+1} = \alpha_j\lambda_j \mathbf{x}_j + \bar{\alpha}_j\bar{\lambda}_j \bar{\mathbf{x}}_j + \beta_I \mathbf{x}'_I + \gamma_I \mathbf{x}''_I \quad \text{sum on j, I} \tag{69}$$

The pseudo energy at time steps n and $n + r$, before and after the group B update, is calculated by substituting (68) and (69) into (38) and with the orthogonality properties from (66–67) reduces to

$$E^n = \left(\alpha_j + \bar{\alpha}_j\right)^2 + \gamma_I\gamma_p \mathbf{z}_{AI}^{\mathrm{T}}\mathbf{M}_A\mathbf{z}_{Ap} + \left(\frac{\mathrm{i}\alpha_j}{\omega_{Bj}} - \frac{\mathrm{i}\bar{\alpha}_j}{\omega_{Bj}}\right)^2 + \beta_I\beta_p \mathbf{z}_{AI}^{\mathrm{T}}(\mathbf{K}'_{AA})\mathbf{z}_{Ap} \tag{70}$$

$$E^{n+r} = \left(\alpha_j\lambda_j + \bar{\alpha}_j\bar{\lambda}_j\right)^2 + \gamma_I\gamma_p \mathbf{z}_{AI}^{\mathrm{T}}\mathbf{M}_A\mathbf{z}_{Ap} + \left(\frac{\mathrm{i}\alpha_j\lambda_j}{\omega_{Bj}} - \frac{\mathrm{i}\bar{\alpha}_j\bar{\lambda}_j}{\omega_{Bj}}\right)^2 + \beta_I\beta_p \mathbf{z}_{AI}^{\mathrm{T}}(\mathbf{K}'_{AA})\mathbf{z}_{Ap} \tag{71}$$

where

$$\mathbf{K}'_{AA} = \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA} \ . \tag{72}$$

It can be shown from (64) that $|\lambda_{Bj}| = 1$ and therefore can be expressed as

$$\lambda_{Bj} = \cos\theta_{Bj} + \mathrm{i}\sin\theta_{Bj} \tag{73}$$

and using the above equation together with α_I given by (33), both (70) and (71) can be simplified to give

$$E^n = E^{n+r} = 4A_j^2 + \gamma_I\gamma_p \mathbf{z}_{AI}^{\mathrm{T}}\mathbf{M}_A\mathbf{z}_{Ap} + \beta_I\beta_p \mathbf{z}_{AI}^{\mathrm{T}}(\mathbf{K}'_{AA})\mathbf{z}_{Ap} \ . \tag{74}$$

Thus, the pseudo energy is also unchanged by the group B update.

Since the multi-time step algorithm consists of a series of group A updates and group B updates, this stability analysis leads to the result that E will always be constant which implies that v and a must always remain bounded for all time. This algorithm is also unconditionally stable since the stability analysis is valid independent of the time steps. In addition, it is interesting to note that the same psuedo energy is conserved for both single time step algorithm and the multi-time step algorithm. This was not the case for the explicit multi-time step algorithm (Smolinski, Sleith and Belytschko 1996) where different psuedo energies were defined for the single and multi-time step schemes.

4 Numerical results

Several example problems were solved using the multi-time step algorithm to numerically verify the previously derived stability analysis and also to examine the accuracy of the scheme.

The example problems consisted of two different beam models composed of C^1 elements as shown in Figs. 2 and 3. For problem 1 the initial displacement is given by $y = 9.76 \times 10^{-5}(24x^2 - x^3)$ and for problem 2 is $y = 1.95 \times 10^{-4}(96x^2 - 16x^3)$ for $0 \le x \le 4$ and is symmetric about the center of the beam. For both problems the density and elastic modulus were taken as, $\rho = 8000$ kg/m^3 and $E = 210$ Gpa, and the cross-sectional area and moment of inertia were, $A = 0.012$ m^2 and

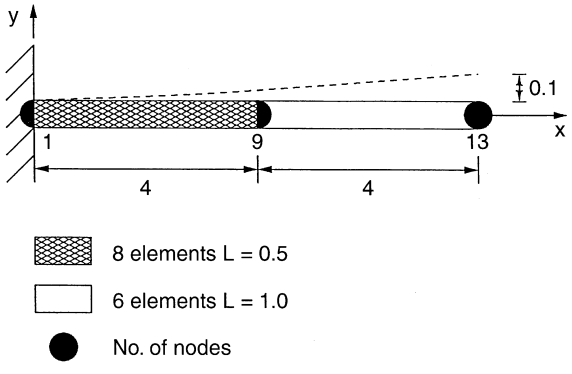

Fig. 2. Problem statement for problem 1

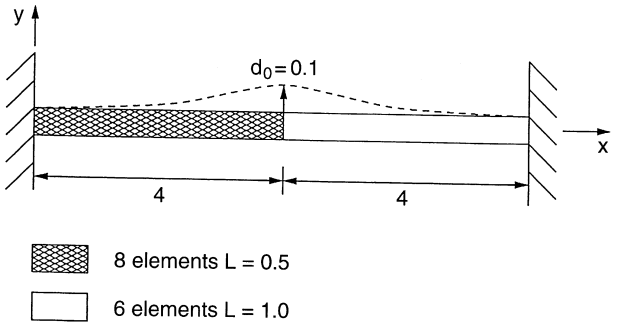

Fig. 3. Problem statement for problem 2

$I = 1.44 \times 10^{-5}$ m^4, respectively. The different time integration cases that were examined for each problem are given in Tables 1 and 2. The time steps for the nodal groups are given in terms of the critical time step for the central difference method as determined by

$$\Delta t_c = \frac{2}{\omega^e_{\max}} \tag{75}$$

where $\omega^e_{\max}$ is the maximum element frequency. Standard trapezoidal rule integration was used for a benchmark and cases of the multi-time step method where the same time step is used for both nodal groups is also considered.

For problem 1 the tip velocity and energy, (38), history are shown in figure 4 for the four cases. From these results it can be seen that the psuedo energy does not change as predicted by the stability analysis. For the velocity there is good agreement between cases 1, 2 and 4, however, case 3 showns considerable error. Comparing case 2, 3 and 4, it seems that a significant error is introduced when there is a large time step ratio between the nodal groups and the total cycle time is also large.

Table 1. Time integration parameters for problem 1

Method	Group A	Group B	Total cycle
node	1–9	10–13	
case 1: standard	$8\Delta t_c$	$8\Delta t_c$	$8\Delta t_c$
case 2: multi-time	$.25\Delta t_c$	$4\Delta t_c$	$8\Delta t_c$
case 3: multi-time	$1.5\Delta t_c$	$6\Delta t_c$	$12\Delta t_c$
case 4: multi-time	$6\Delta t_c$	$6\Delta t_c$	$12\Delta t_c$
Δt_c	$\Delta t_c = 20.3 \times 10^{-5}$	$4\Delta t_c$	

Table 2. Time integration parameters for problem 2

Method	Group A	Group B	Total cycle
node	1–9	10–13	
case 1: standard	$8\Delta t_c$	$8\Delta t_c$	$8\Delta t_c$
case 2: multi-time	$0.25\Delta t_c$	$6\Delta t_c$	$12\Delta t_c$
case 3: multi-time	$4\Delta t_c$	$8\Delta t_c$	$16\Delta t_c$
case 4: multi-time	$8\Delta t_c$	$8\Delta t_c$	$16\Delta t_c$
Δt_c	$\Delta t_c = 20.3 \times 10^{-5}$	$4\Delta t_c$	

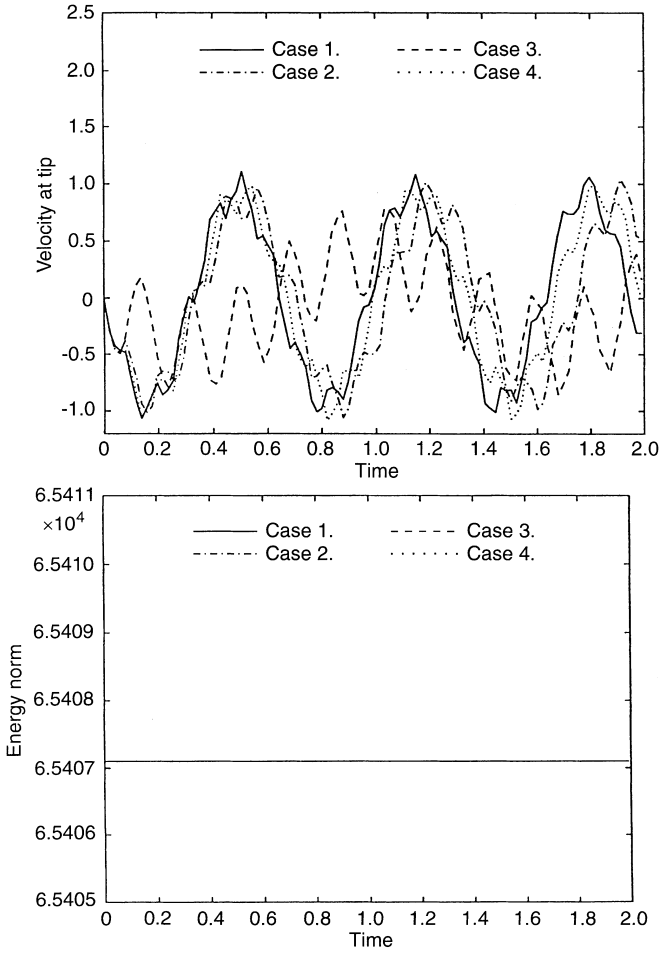

Fig. 4. Numerical results for problem 1

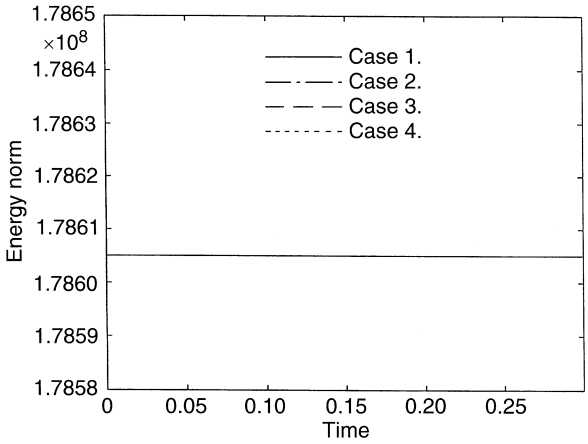

Fig. 5. Energy time history for problem 2

In Fig. 5 the psuedo energy as a function of time is shown for the four cases of problem 2. Again, this energy remains constant which supports the derived stability analysis. The velocity time histories for two locations in the beam are given in Fig. 6. There is reasonable agreement between cases 1, 2 and 4 with case 3 displaying the largest

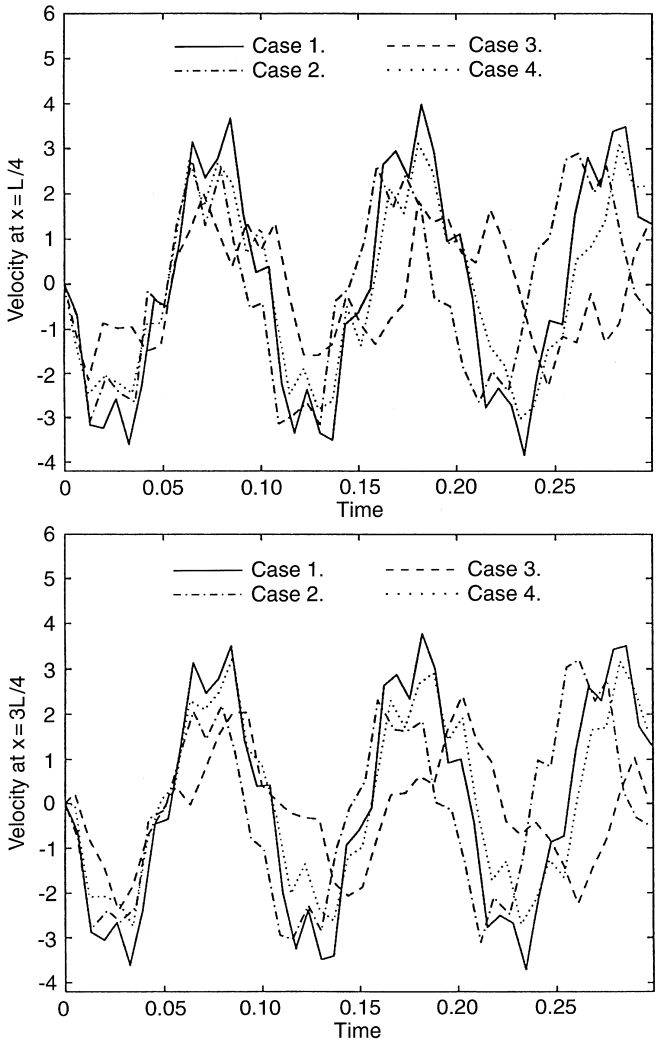

Fig. 6. Velocity time history at two locations for problem 2

error. Again, it appears that for larger total cycle times the larger the difference between the group time steps the greater the error.

5
Conclusions

An implicit subcycling time integration method has been developed based on the trapezoidal rule integration algorithm. Using an energy analysis, the method has been shown to be unconditionally stable and conserves the same psuedo-energy as the standard algorithm. Numerical tests have shown that the accuracy of the method can degrade if the total cycle time or the time step ratio becomes too large. These effects may be the result of the locally implicit nature of the algorithm in that information can only flow between nodes in the nodal group being updated and not throughout the entire mesh as with the standard algorithm. Further work is being done to determine appropriate criteria for the selection of time steps.

References

Belytschko T (1980) Partitioned and adaptive algorithms for explicit time integration, In: Nonlinear Finite Element Analysis in Structural Mechanics, W. Wunderlich, et al. Eds 572–584, Springer-Verlag

Belytschko T, Hughes TJR (1983) Computational Methods for Transient Analysis. North Holland, Amsterdam

Belytschko T, Mullen R (1977) Explicit integration of structural problems, In: Finite Elements in Nonlinear Mechanics, P. Bergen, et al. Eds Vol. 2:697–720

Belytschko T, Lu YY (1993) Explicit multi-time step integration for first and second order finite element semidiscretizations, Comp. Meth. Appl. Mech. Engrg. 108:353–383

Belytschko T (1996) Private communication

Hughes TJR, Liu WK (1978) Implicit-explicit finite elements in transient analysis: stability theory. J. Appl. Mech. 45:371–374

Liu WK, Belytschko T (1982) Mixed-time implicit-explicit finite elements for transient analysis. Comp. Struct. 15:445–450

Liu WK, Lin JI (1982) Stability of mixed-time integration schemes for transient thermal analysis. Int. J. Numer. Meths. Heat Trans. 5:211–222

Liu WK (1983) Development of mixed time partition procedures for thermal analysis of structures. Int. J. Numer. Meths. Engrg. 19:125–140

Neal MO, Belytschko T (1989) Explicit-explicit subcycling with non-integer time step ratios for structural dynamics systems. Comp. Struct. 31(6):871–880

Park KC (1980) Partitioned transient analysis procedures for coupled field problems: stability analysis. J. Appl. Mech. 47:370–376

Smolinski P, Sleith S, Belytschko T (1996) Stability of an explicit multi-time step integration algorithm for linear structural dynamics equations. Comp. Mech. 18(3):236–244