Implementation and application of elastodynamic boundary element discretizations with improved stability properties

Received October 1995
Accepted July 1996

E. Siebrits
Schlumberger Dowell, Tulsa, OK, USA, and
A.P. Peirce
Department of Mathematics, University of British Columbia, Vancouver, BC, Canada

Introduction

There is growing evidence of intermittent numerical instabilities in boundary integral elastodynamic models (Andrews, 1994; Mack, 1991; Siebrits, 1992). We have used the term "intermittent instabilities" because of the way in which the instabilities appear and disappear as the time step is changed. As an example of this type of instability, consider a fixed spatial discretization of a given elastodynamic problem, and allow the time step to change. The time domain boundary element model can be unstable for a certain time step and become stable if the time step is increased. If the time step is increased further, then the boundary element model may become unstable again. In addition, these instabilities may occur for certain problems and not for others depending on the specific geometry of the problem, i.e. the particular combination of spatial modes that are active in the problem. This intermittent instability is clearly unacceptable, as one cannot provide coherent guidelines about the appropriate choice of meshing parameters.

In this article, we will restrict ourselves to the space-time formulation of the boundary element equations. We briefly outline the various boundary element formulations. We show how the indirect boundary element methods can be obtained from the direct boundary element formulation. We also outline the various temporal and spatial discretization procedures as well as a strategy for time marching the resulting system of algebraic equations. In order to illustrate the nature of the instabilities and how pervasive they are, we provide examples of instabilities for the direct formulation, and both the fictitious stress and displacement discontinuity indirect formulations.

The authors acknowledge the support of the Safety in Mines Research Advisory Committee of the South African Department of Mineral and Energy Affairs. The second author also acknowledges the support of the National Science and Engineering Research Council of Canada.

Engineering Computations, Vol. 14 No. 6, 1997, pp. 669-695.

We outline the practical implementation of various time stepping schemes in boundary element algorithms. In particular we discuss the implementation of schemes which assume that the unknown field is constant in time – which is suitable for the direct and fictitious stress methods. Numerical results are given for the constant in time schemes for which different collocation points are used and their stability properties are contrasted. Next we detail the practical implementation of two new time stepping schemes for linear fields in time, namely the ε-scheme and the half-step scheme. The enhanced stability characteristics of these new schemes are clearly demonstrated by comparing them with standard time stepping schemes commonly used in boundary element algorithms.

Boundary element formulations

Boundary integral formulations

The direct boundary element method is well documented (Kobayashi, 1985) and the derivation will not be repeated here. The direct boundary element method equations are obtained by combining the dynamic reciprocal theorem of Betti-Rayleigh (Love, 1944) with the appropriate fundamental solutions (Eringen and Suhubi, 1975). In the absence of body forces and given zero initial conditions, the direct boundary element equations are given by

$$u_k(\underline{\xi},t) = \int_S \left[U_{ik}(\underline{x},t;\underline{\xi},0) * t_i(\underline{x},t) - T_{ikj}(\underline{x},t;\underline{\xi},0) n_j * u_i(\underline{x},t) \right] dS(\underline{x}) \qquad (1)$$

where $U_{ik} = T_{ikj}$ are the fundamental displacements and stresses, respectively, due to a unit impulsive load, and "*" is the time convolution operator (Eringen and Suhubi, 1975).

The indirect boundary element methods (i.e. the fictitious stress method and the displacement discontinuity method) can be obtained from the direct boundary element method by adding an interior to an exterior domain problem (Loken, 1992; Siebrits, 1992; Tian, 1990). By subtracting the equations for the interior region from those for the exterior region, an equation similar to (1) is obtained in which the tractions t_i and displacements u_i are replaced by the jumps in traction and displacement across the boundary between the two regions. By requiring that the displacement jumps are zero across the interface, we obtain the fictitious stress method. The displacements and stresses for the fictitious stress method are given by

$$\begin{aligned} u_k(\underline{x},t) &= \int_s U_{kl}(\underline{x},t;\underline{\xi},0) * F_i(\underline{\xi},t) dS(\underline{\xi}) \\ \sigma_{kj}(\underline{x},t) &= \int_s T_{kji}(\underline{x},t;\underline{\xi},0) * F_i(\underline{\xi},t) n_j dS(\underline{\xi}) \end{aligned} \qquad (2)$$

where $F_i = t_i^+ - t_i^-$ are the traction jumps across the fictitious stress surface S.

Similarly, by requiring that the tractions are continuous across the interface, we obtain the displacement discontinuity method. The displacement and stress equations for the displacement discontinuity method are given by

$$u_k(\underline{x},t) = \int_S T_{kij}(\underline{x},t;\underline{\xi},0)n_j * D_i(\underline{\xi},t)dS(\underline{\xi})$$
$$\sigma_{kl}(\underline{x},t) = \int_S S_{klij}(\underline{x},t;\underline{\xi},0)n_j * D_i(\underline{\xi},t)dS(\underline{\xi}) \quad (3)$$

where $D_i = u_i^+ - u_i^-$ are the displacement jumps across the displacement discontinuity surface S, and S_{klij} is given by

$$S_{klij} = -\left[\lambda \frac{\partial T_{mij}}{\partial \xi_m}\delta_{kl} + \mu\left(\frac{\partial T_{kij}}{\partial \xi_l} + \frac{\partial T_{lij}}{\partial \xi_k}\right)\right] \quad (4)$$

where λ and μ are the Lame constants, and δ_{kl} is the Dirac delta function.

Discretization of boundary integrals

The boundary integrals in the above equations contain two types of integrals, viz. time and space. The time integrals (embodied in the time convolution operator) are discretized into time steps, with a particular functional variation over each time step (e.g. constant, linear).

The spatial boundaries are also discretized into elements. Each element is assumed to have certain geometric properties (e.g. straight or curved elements) and the functional variation over each element is assumed to be of a particular order (e.g., constant, linear, quadratic).

The temporal integrals can all be performed analytically, and this is well documented (Loken, 1992; Mack, 1991; Siebrits, 1992; Tian, 1990). The spatial integrations are often determined numerically (Banerjee, *et al.*, 1987) especially in the case of higher order geometrical and functional variations over each element. In the case where each element is assumed to be straight (or flat), these integrals can also be determined analytically (Loken, 1992; Mack, 1991; Siebrits, 1992; Tian, 1990).

There are restrictions governing the choice of functional variation in space and time. For example, in the displacement discontinuity method, a piecewise constant functional variation in time (in two and three dimensions) is not possible because it leads to singular integrated stress expressions (Loken, 1992; Mack, 1991; Siebrits, 1992). Hence, a minimum requirement of the displacement discontinuity method is a linear variation within each time step, with continuity between time steps. Furthermore, stability conditions govern which orders of time and space functional variations are permissible (Peirce and Siebrits, 1995).

Temporal integrations

Assuming a piecewise linear time variation for the approximating function, denoted by $f(t)$, the integrations can be performed for the special case $f(t) = t/\Delta t$, and then generalized by combining three such staggered functions to obtain a "hat" function, from which the piecewise linear time variation can be constructed. Hence,

$$f(t) = H(t - t_{k-1})\frac{\tau_{k-1}}{\Delta t} - 2H(t - t_k)\frac{\tau_k}{\Delta t} + H(t - t_{k+1})\frac{\tau_{k+1}}{\Delta t} \quad (5)$$

where $t_k = k\Delta t$ and $\tau_k = t - t_k$. The above approach is possible because of the linearity of the boundary element method (so that the principle of superposition applies), and because of the time translation property of all the fundamental solutions. Combining a hat function at each time step results in a piecewise linear temporal variation (Loken, 1992; Mack, 1991; Siebrits, 1992; Tian, 1990). Piecewise constant time elements can be constructed in a similar way, by combining three sets of Heaviside functions appropriately. Higher order in time variations can also be constructed, but are not covered here.

Spatial integrations

In the two-dimensional case, if we assume that the elements are straight line segments, then analytical integrations are possible, for functional variations that are constant, linear or quadratic along the element. This has been fully covered elsewhere (Siebrits, 1992; Tian, 1990) and will not be repeated here. In the three-dimensional case, if we assume that each element is flat, then analytical integrations are once again possible (Loken, 1992; Mack, 1991). Alternatively, numerical integration procedures can be used. Care must be taken, especially in the displacement discontinuity element method, to avoid numerical problems with singularities at element edges because of the hypersingularity of the fundamental solutions of this method. In fact, the hypersingular nature of the displacement discontinuity method stress equations precludes the use of numerical integration schemes. Furthermore, element integrations must be causal (i.e. partial integrations are performed for those portions of the elements that fall within the causal space-time "light cone") in order to ensure accurate influence coefficient calculations.

Numerical implementation

The discretization of the time and space integrals in any of the time domain direct or indirect boundary element methods ultimately leads to a system of time marching algebraic equations of the form

$$\underline{b}_m = \underline{\underline{C}}_0 \underline{F}_m + \sum_{k=1}^{m-1} \underline{\underline{C}}_{m-k} \underline{F}_k \quad (6)$$

where

$\underline{F}$ = vector of unknown boundary tractions and/or displacements, or fictitious stresses, or displacement discontinuities

$\underline{\underline{C}}$ = influence coefficient matrix

$\underline{b}$ = boundary displacement and/or traction vector

m = current time step number.

Introduction of (5) into (6) results in a time stepping algorithm used in most current boundary element schemes with linear in time functional variations,

which we term the standard or Trapezoidal scheme, whose time basis functions are depicted in Figure 11.

The matrices $\underline{\underline{C}}$ are fully populated in general. It is clear that the unknown quantities $\underline{F}$ at the current time step m are obtained via a convolution between the known coefficients and known quantities from all previous times (in the two-dimensional case).

Algorithm (6) can be explicit ($\underline{\underline{C}}_0$ diagonal) or implicit ($\underline{\underline{C}}_0$ not diagonal), depending on the type of discretization that is used. If the functional variation is constant across an element, then the algorithm can be made explicit by choosing a small enough time step, given by $Q1 = c_1 \Delta t/\Delta x < 0.5$, where Δx is the element size. In other words, the system is explicit if the compressional wave front travels less than half an element length in one time step. Of course, this presupposes that neighbouring elements are not at acute angles to each other, or closer than half an element length. In such cases, the time step restriction is more severe. If the functional variation over each element is variable, then the algorithm will always be implicit, because collocation points within each element ensure that cross-coupling occurs regardless of the size of the time step, and hence matrix $\underline{\underline{C}}_0$ is never diagonal.

The stability of the algorithm is not guaranteed if the time step is chosen such that the scheme becomes implicit. This has to be tested, and will be shown later to depend on the functional variations across time and space elements and the geometric distribution of elements.

Evidence of instabilities

Mack (1991) and Siebrits (1992) have both noted numerical instabilities in their three-dimensional (3D) and two-dimensional (TWO4D) displacement discontinuity codes, respectively. 3D uses linear in time and constant in space functional variation on flat rectangular elements. TWO4D uses linear in time, and either constant in space ("constant/linear" scheme) or linear in space ("linear/linear" scheme) functional variations on straight-line elements. Tian (1990) and Loken (1975) have both also noted numerical instabilities in their two-(IBEM2) and three-dimensional (3DFS) fictitious stress codes, respectively. 3DFS uses linear in time and constant in space functional variations on flat rectangular elements. IBEM2 uses constant or linear in time and constant in space functional variations on straight elements. Tian's (1990) direct boundary element code (DBEM2), which uses linear in time and quadratic in space functional variations, also exhibits numerical instabilities.

The above codes all use analytical integrations in time and space. A more recently published direct boundary element code, QUADPLET, (Dominguez, 1993) which uses quadratic spatial and linear temporal elements, and numerical integrations for the spatial integrals, also goes unstable. For example, a cavity, where the circumference is loaded suddenly by a normal traction axisymmetrically (Selberg, 1951), is clearly unstable by 2,000 time steps, as shown in Figure 1. In this problem, 16 elements have been used around the cavity boundary, and $Q1 = 0.3$. The choice of loading configuration can be designed to

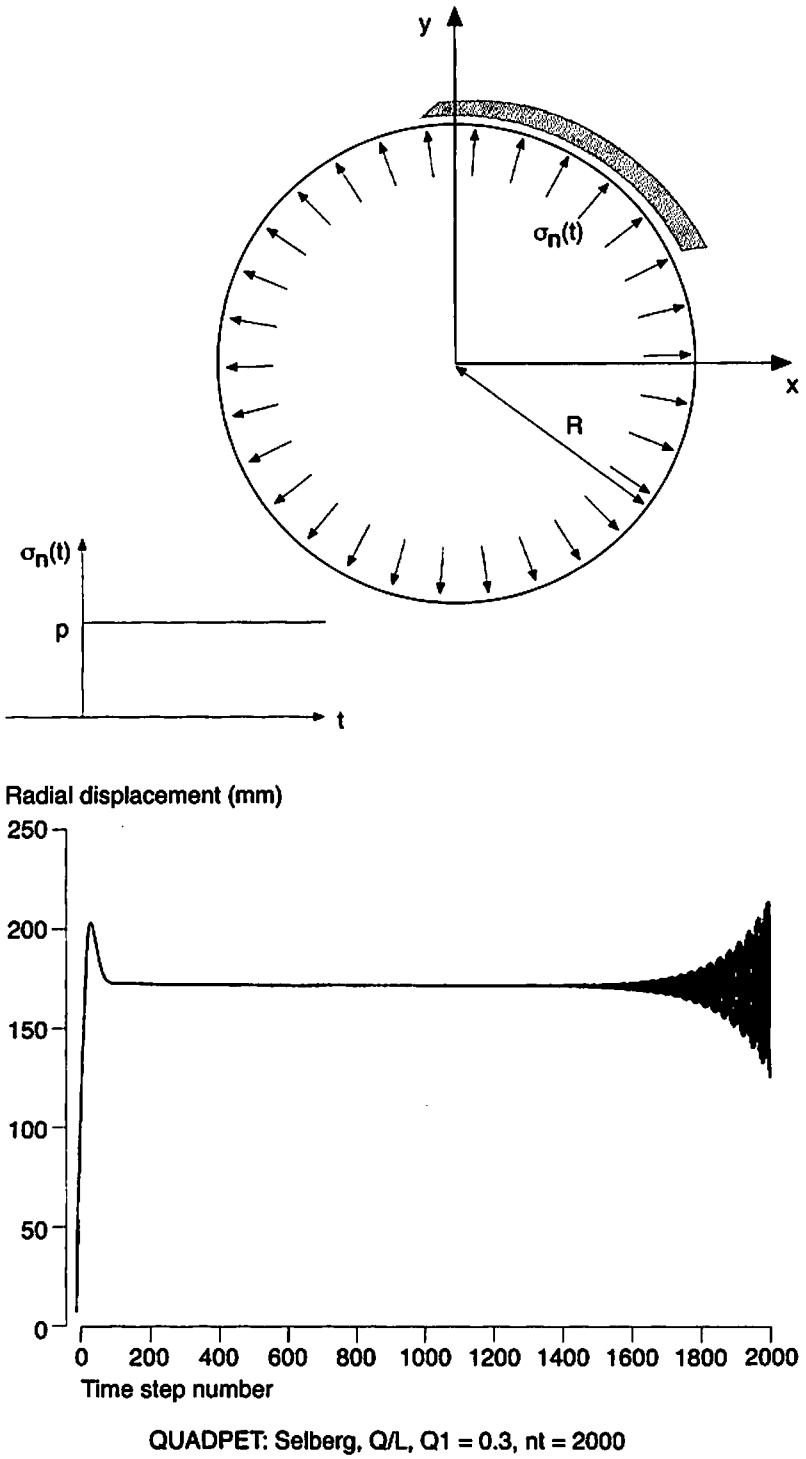


Figure 1.
Selberg's suddenly pressurized cavity problem which exhibits instabilities

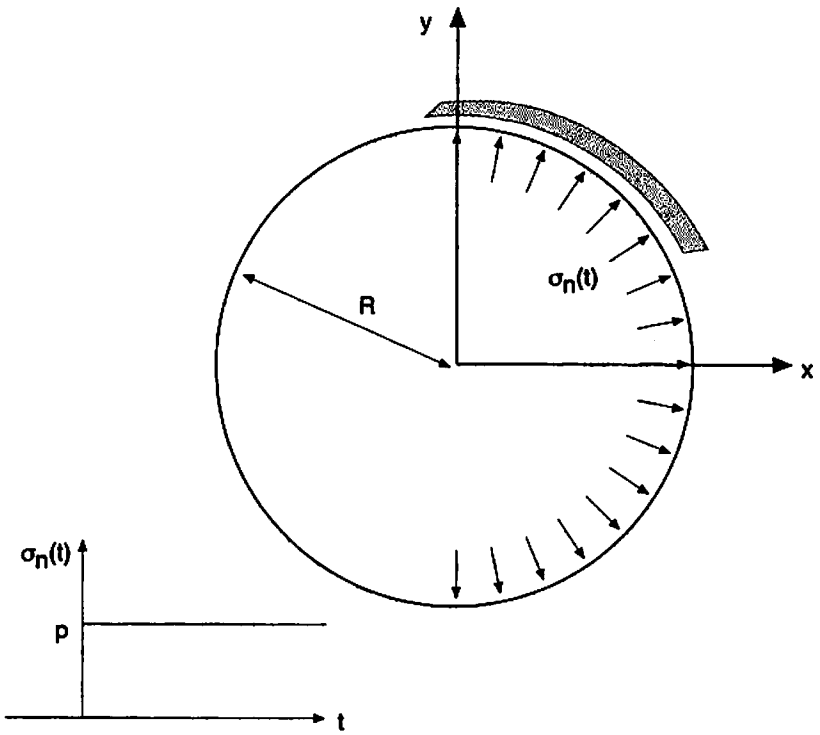
ensure that the instability becomes physically evident sooner rather than later. If the cavity is loaded on half of the circumference, then the instability emerges by 300 time steps, as shown in Figure 2, although this is a shear component of displacement (in the previous example, only the radial displacement is non-zero).

There are also other independent references made to unstable boundary element methods in the literature (Andrews, 1994; Koller *et al.*, 1992). Koller *et al.* (1992) developed a boundary integral method to model antiplane automatic dynamic crack growth. Their method produces spurious numerical "oscillations" which increase with time, and they claim that these oscillations occur with respect to the spatial domain, and not the time domain. This they confirm by implementing a static version of the traction boundary integrals using the same spatial discretization (viz. collocation at midpoints of elements) and note that the same kinds of oscillations arise. We shall see later that the numerical instabilities in boundary element methods are a coupled spatial and temporal phenomenon. Koller *et al.* (1992) also claim to eliminate their oscillations by reformulating the original problem as a penalized least-squares problem, using a stabilizing functional in the sense of Tikhonov and Arsenin (1977). They claim that the numerical oscillations disappear by making appropriate choices of the weighting factors, but do not validate this by numerical example or analytical study.

Andrews (1994) modelled mixed-mode shear slip with a boundary integral approach, where the spatial convolutions were performed in the Fourier domain. He also notes the presence of "oscillations" which grow worse with time, and comes to the (sad) conclusion that he would use finite difference methods if he were to redo his work.

Numerical instabilities do not necessarily disappear if the time step is reduced – they are intermittent in nature, and problem dependent. The simplest TWO4D problem that illustrates this is an antiplane strain constant/linear two element problem, where the elements are located opposite each other, $h = 2\Delta x$ apart, as shown in Figure 3. Table I highlights the intermittent nature of the numerical instability that is observed in this case. In Table I, $Q2 = c_2\Delta t/\Delta x$. The loading pattern does not determine whether an instability will occur. However, it does affect how soon it will manifest itself. The intermittent nature of the instabilities can be traced directly to cases where wave fronts intersect elements. This will be covered in more detail later in this section. Depending on the particular combination of time step, spatial step and problem geometry, numerical oscillations can: grow exponentially as a classical numerical instability; grow non-exponentially as a resonant instability; beat, with successive beats growing, decaying or remaining unchanged, or merely oscillate with a decrease in amplitude or an unchanging amplitude. In other words, it is possible to generate almost any type of oscillation imaginable.

There are numerous ways of delaying the onset of visible numerical instabilities. For example, averaging techniques from time step to time step, and even repetitive averaging techniques, have been used (Manolis, *et al.*, 1986). Repetitive averaging is not desirable because incorrect solution paths can be followed. The use of double precision for the kernel generation and summation



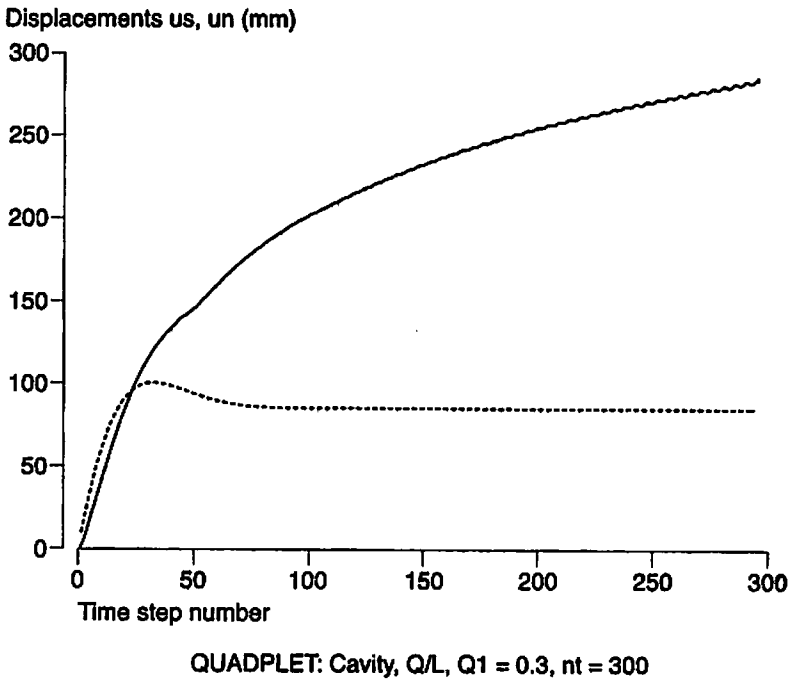


Figure 2.
Cavity problem, with loading over half of circumference, which exhibits instabilities

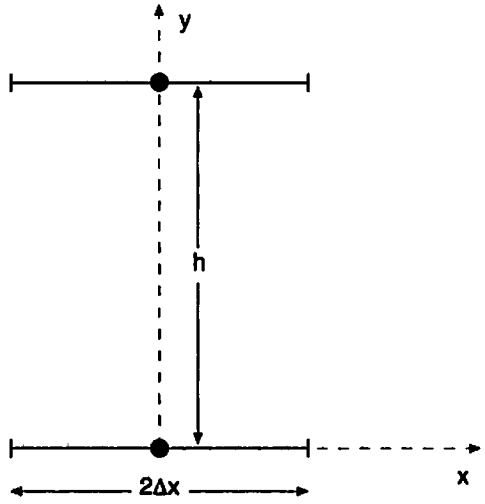


Figure 3.
Two element problem, geometry which exhibits instabilities

routines delays the onset of numerical instabilities. Normalization of the influence coefficients is also beneficial.

The reader might question the necessity of worrying about numerical instabilities at all, especially as they normally appear at later times. However, instabilities can emerge very quickly for problems where the influence coefficient matrix structure is such that the matrices are poorly conditioned (e.g. closely parallel rows of elements). In these cases, the instabilities swamp the transient behaviour at early times. (Closely parallel rows of elements are also difficult to solve with static boundary element codes. In fact, in the two-dimensional elastostatic displacement discontinuity method, with constant functional variation over each element, if any two elements are parallel and exactly half an element length apart, then the problem is unsolvable because the source and receiver influences are identical, resulting in a singular coefficient matrix. In elastodynamics, this problem goes unstable very quickly.)

In the dynamic displacement discontinuity method, even a single row of elements can go unstable. For example, in the constant/linear plane strain version of TWO4D, a single straight line of 20 elements, with $Q1 = 0.5$, is unstable by 500 time steps. In two dimensions, a Rayleigh wave propagating along a free surface does not decay with distance from the source (Graff, 1975) Thus, element to element influences along a mathematical crack construction, such as provided by the displacement discontinuity method, contain a component that does not decay with distance. We will see later that the spatial

Table I.
Evidence of intermittent instability in antiplane strain TWO4D with standard time stepping scheme

$Q2$	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80
Stable?	no	no	yes	no	yes	yes	no	yes
$Q2$	0.90	1.00	1.10	1.20	1.30	1.34	1.40	1.50
Stable?	yes	yes	yes	yes	yes	no	yes	yes

decay rate of influences plays an important role in the stability properties of boundary element (Peirce and Siebrits, 1995).

A problem which has been investigated intensively by the authors, is the so-called "Hook" problem, involving two rows of displacement discontinuity elements with prescribed tractions as boundary conditions. Figure 4 depicts the geometry and loading. Incidentally, this particular problem was the very first problem ever run with TWO4D, and has become a testbed for alternative strategies to cope with numerical instabilities. The problem is somewhat ill-conditioned because of the acute angle between the two rows of elements. Depending on the choice of $Q1$ (or $Q2$ in antiplane strain), this problem may or may not go unstable. This applies to both the constant/linear and linear/linear versions of TWO4D. Figure 5 shows a typical unstable result obtained from the constant/linear antiplane strain version of TWO4D with $Q2 = 0.7$, where the displacement discontinuity history of element 21 (in the corner of the hook) is plotted. The smooth line through the unstable result is the equivalent solution obtained using a novel scheme developed by the authors. This scheme, which is termed the Half-step scheme, is described in a later section. Figure 6 is a plot of the light cones at the first few time steps, for elements 19 and 20. It is clear from

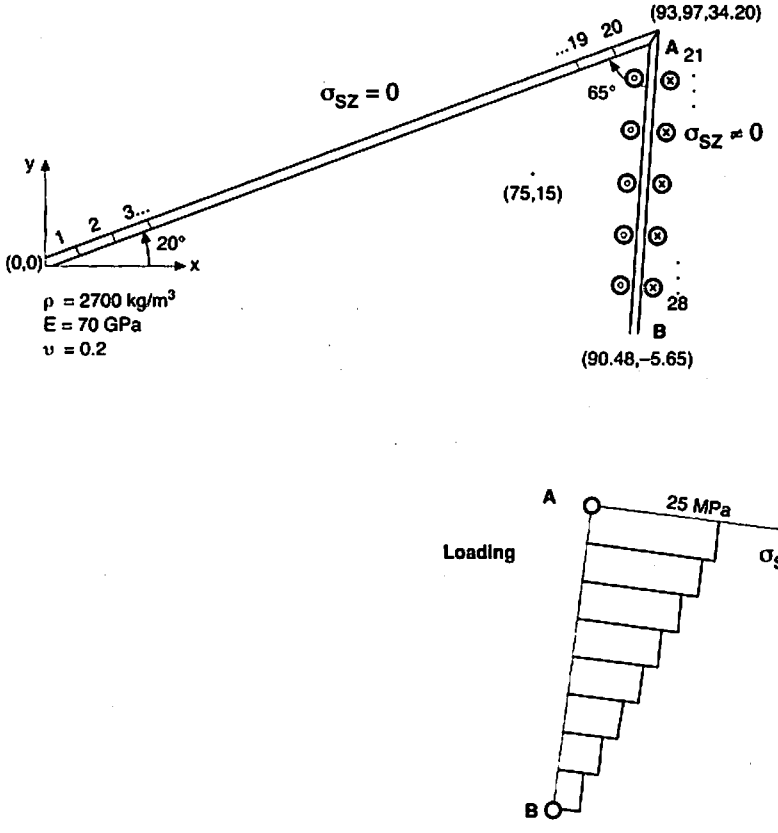


Figure 4.
Antiplane strain Hook problem geometry which exhibits numerical instabilities

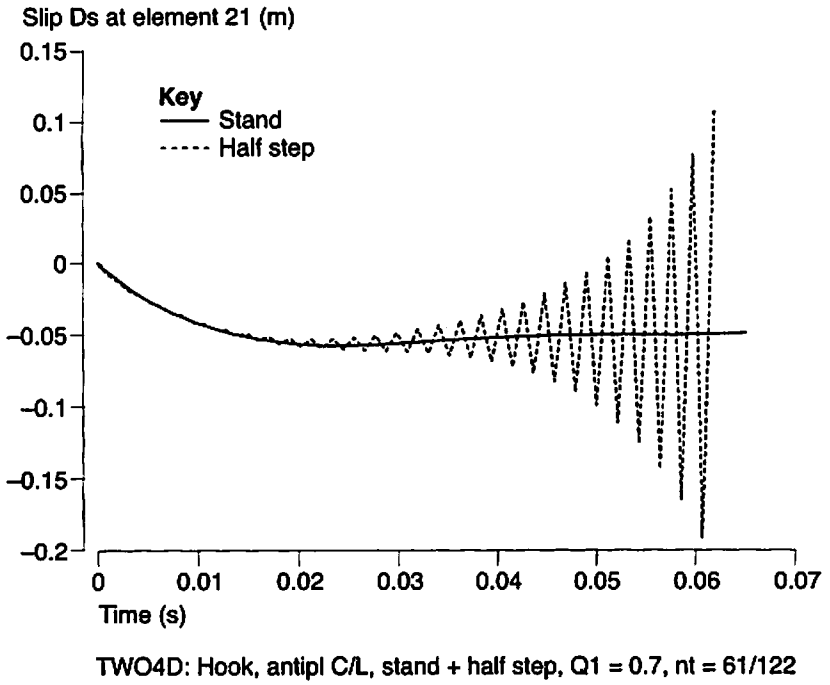


Figure 5. Displacement discontinuity history of element 21 for antiplane strain Hook problem

the figure that the shear wave front from element 19 just chops element 21 at the second time step (shown as a dashed line). We will see later that the influence coefficients (or stress wave fronts) orthogonal to a source element are the largest, and are most likely to cause numerical instabilities. Figure 7 shows the same diagram, but for the case $Q2 = 0.67$. Notice that in this case, the light cone at the second time step between elements 19 and 21 just misses element 21. This run turns out to be stable.

It can be shown (Peirce and Siebrits, 1995) that, in the case of wave fronts intersecting neighbouring elements, the stability of the problem is dominated by the relative magnitude of the effect that the source element has on itself and the effect that it has on its neighbour. Due to diffraction of the pulse caused by a change in the spatial variation of the forcing function, it is possible for the stress effect remote from the source element to be larger than the effect that the element has on itself. This positive feedback between the two elements results in the instability.

Implementation of time stepping schemes

Constant time variation schemes

There are three constant in time schemes, that can be applied to the direct and fictitious stress methods, viz. Backward Euler, Midpoint and Forward Euler schemes. Recall that constant in time functional variations cannot be applied to the displacement discontinuity method because of the hypersingular nature of the kernels. However, these schemes can be implemented in the fictitious stress

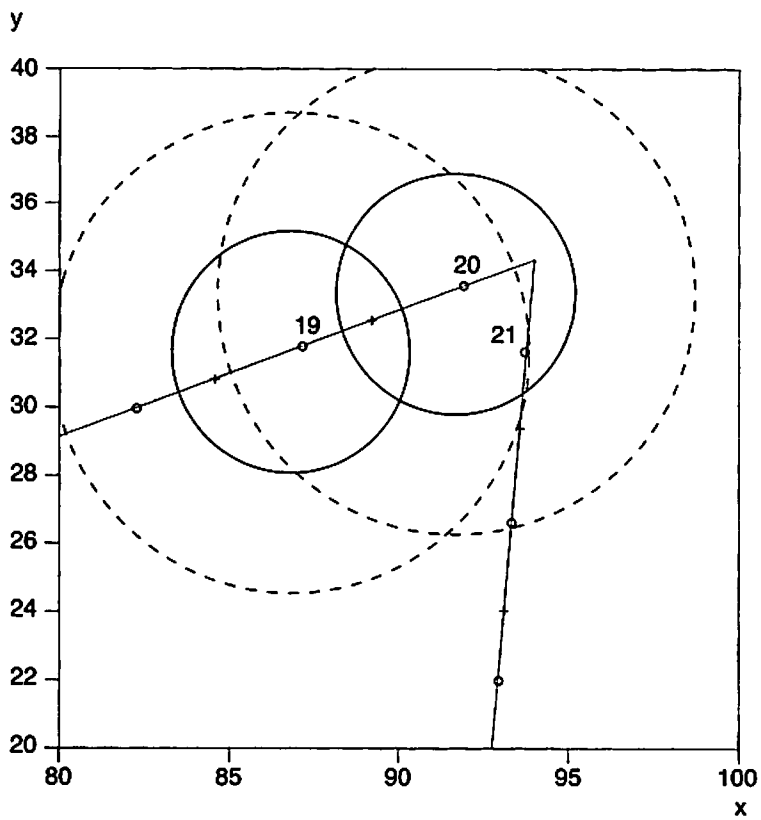


Figure 6.
Light cones for Hook problem with $Q2 = 0.07$

and direct boundary element methods. In the Forward Euler time stepping scheme, the influence coefficients are collocated at the start of each time step, as shown in Figure 8.

In the Backward Euler time stepping scheme, collocation is performed at the end of each time step, as shown in Figure 9. In the Midpoint scheme, the influence coefficients are collocated at the midpoint of each time step, as shown in Figure 10. The Backward Euler scheme implementation is summarized below in the pertinent bit of FORTRAN-77 pseudo code:

```
MAIN
DO 100 K = 1, NTIM   ! time loop
      DO 200 I = 1, NEQU   ! receiver loop
            DO 200 J = 1, NEQU   ! source loop
                  CALL COEFF (I, J, K)
      200 CONTINUE
100 CONTINUE
CALL SOLVE
END
```

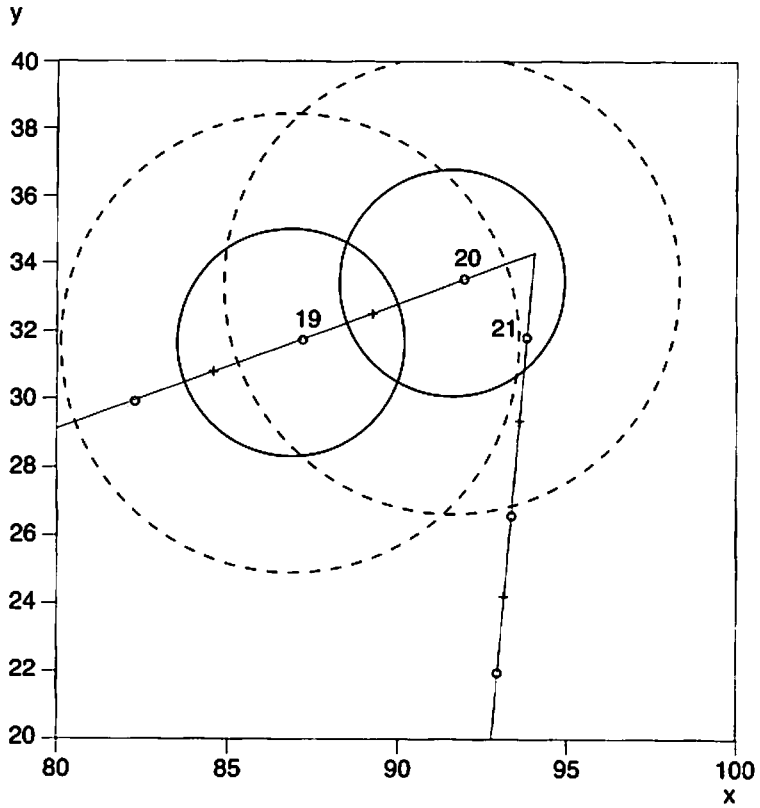


Figure 7.
Light cones for Hook problem with $Q2 = 0.67$

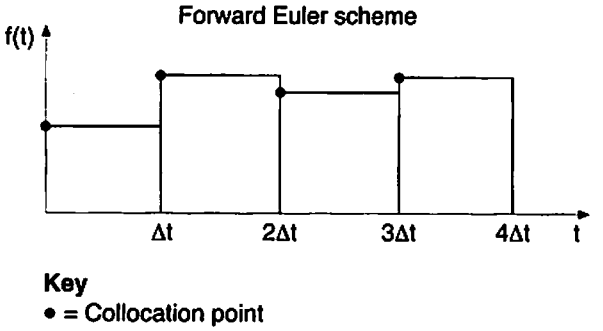


Figure 8.
Basis functions for the Forward Euler scheme

```
SUBROUTINE COEFF (I, J, K)
DELT = QI*DELX/CI
TIME = K*DELT
C(I, J, K) = ...
RETURN
```

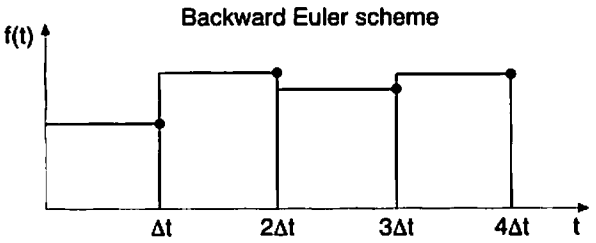


Figure 9.
Basis functions for the Backward Euler scheme

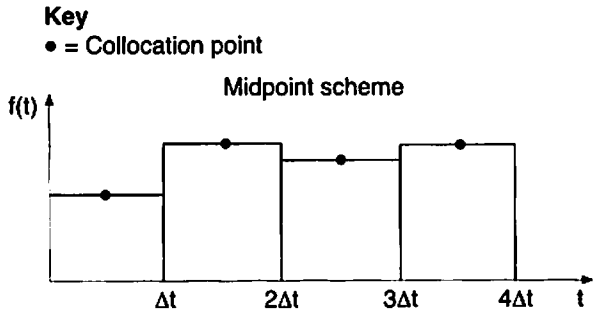


Figure 10.
Basis functions for the Midpoint Euler scheme

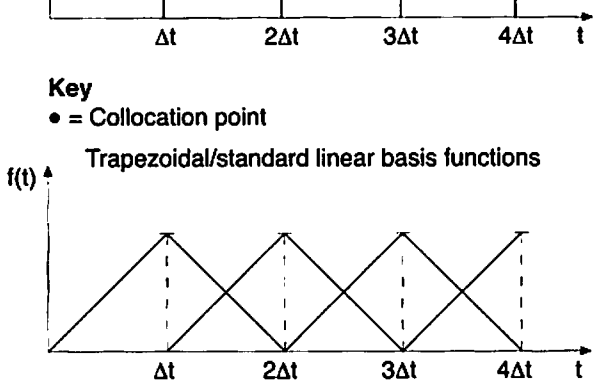


Figure 11.
Basis functions for the Trapezoidal/Standard linear scheme

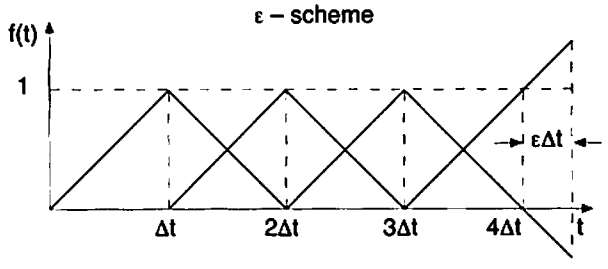


Figure 12.
Basis functions for the ε-scheme

where QI = dimensionless factor $c_i\Delta t/\Delta x$, $i = 1$ if plane strain, or 2 if antiplane strain. In order to change this scheme into a Midpoint scheme, we simply rewrite the above coefficient module as:

```
SUBROUTINE COEFF (I, J, K)
DELT = QI*DELX/CI
```

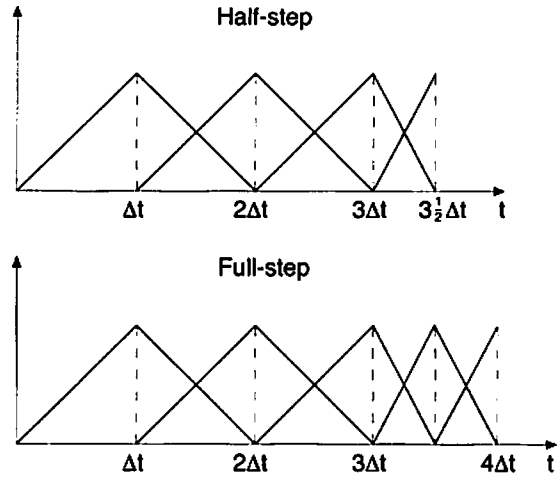


Figure 13.
Basis functions for the Half-step scheme

```
TIME = K*DELT-0.5*DELT
C(I, J, K) = ...
RETURN
```

The coefficient module of the Forward Euler scheme is given by:

```
SUBROUTINE COEFF (I, J, K)
DELT = QI*DELX/CI
```

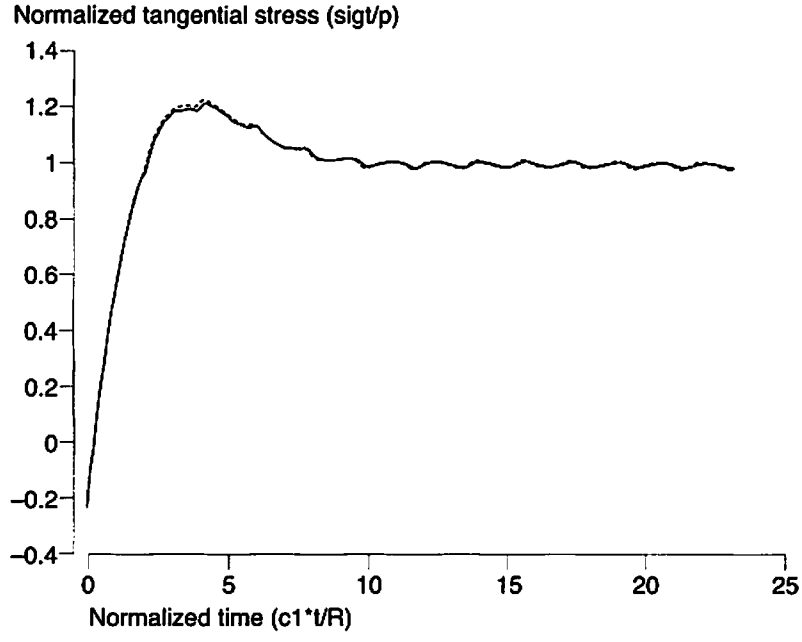


Figure 14.
Comparison between Midpoint constant in time and Trapezoidal linear in time schemes

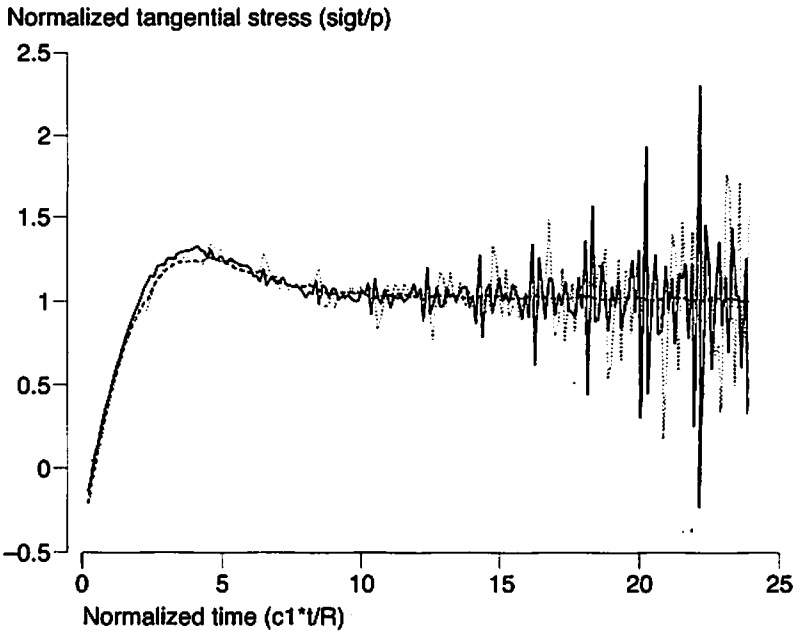


Figure 15.
Constant in time Backward and Forward Euler, and Midpoint schemes

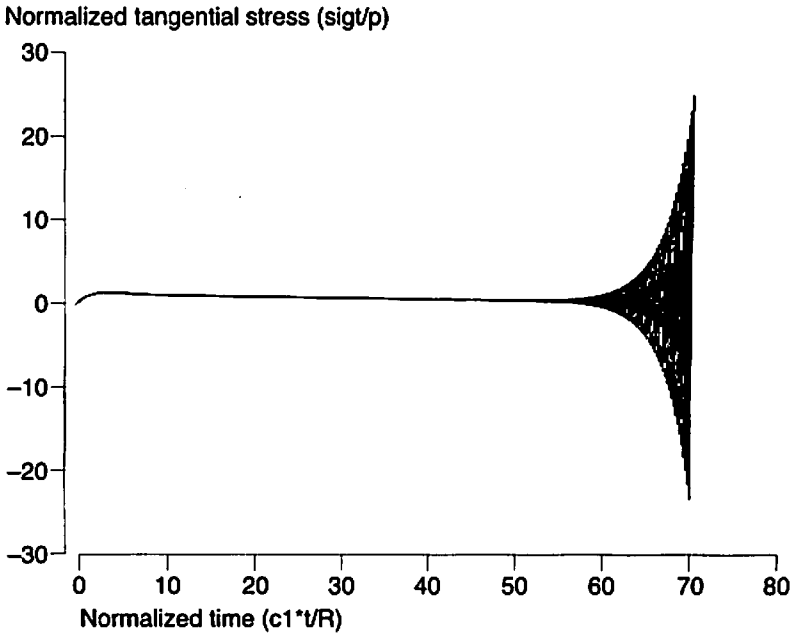


Figure 16.
Fictitious stress Midpoint scheme showing instability

```
TIME = K*DELT-DELT
C(I, J, K) = ...
RETURN
```

The Midpoint scheme is more accurate and more stable than the Backward or Forward Euler schemes. Figure 14 shows the tangential stresses calculated in the Selberg problem by a Trapezoidal linear in time versus a Midpoint constant in time fictitious stress formulation. The results are almost identical. The constant in time Backward and Forward Euler schemes give less accurate results, as shown in Figure 15. The Midpoint scheme does go unstable eventually, as shown in Figure 16.

ε-scheme

It can be shown (Peirce and Siebrits, 1995) that stability is enhanced by devising a time stepping scheme that places more weight on the last unknown $\underline{F}_m$ in (6), or equivalently, on the self-effect terms in $\underline{C}_0$. This can be achieved by integrating the function $f(t)$ in (5) over the time interval $(0, m\Delta t + \varepsilon\Delta t)$ instead of the usual $(0, m\Delta t)$. This has the net effect of improving the stability of the time stepping algorithm, but does introduce a slight shift in the results, which increases with increasing ε. For $\varepsilon \approx 1$, significant errors are observed, whereas for $\varepsilon \approx 0$, the errors are small. Unfortunately, stability improves with increasing ε, and hence, with an accompanied loss in accuracy. Figure 12 depicts the time basis functions for this scheme.

The ε-scheme is trivial to implement in any time domain boundary element scheme. Assuming that the code is structured in a modular fashion, the changes amount to two lines of coding. The following abbreviated code summarizes the implementation of the Trapezoidal time marching algorithm:

```
SUBROUTINE COEFF (I, J, K)
DELT = QI*DELX/CI
TIME = K*DELT
C(I, J, K) = ...
RETURN
```

The ε-scheme is identical to the Trapezoidal one, except that, in module COEFF, we make the following changes:

```
SUBROUTINE COEFF (I, J, K)
DELT = QI*DELX/CI
TIME = (K+EPS)*DELT
C(I, J, K) = ...
RETURN
```

where EPS is a small factor by which the region of integration in time (i.e. total time of integration) is adjusted. Notice that, if EPS = 0.0, then we recover the Trapezoidal time stepping scheme. Figure 12 shows the changes to the temporal hat functions made by the introduction of the ε-scheme compared with the Trapezoidal scheme. Computer run times and storage are unaffected by the introduction of the ε-scheme.

In order to demonstrate the ε-scheme, we consider the plane strain version of the Hook problem (as depicted in Figure 4 for the antiplane strain case). Figure 17 shows results from both the Trapezoidal and ε-schemes, for the case $\varepsilon = 0.1$. The ε-scheme has improved stability characteristics, but not to a significant extent.

Half-step scheme

An alternative scheme to the ε-scheme is the so-called "Half-step" scheme, which was devised (Peirce and Siebrits, 1995) to increase the size of the self-effect without a loss in accuracy. This was achieved by ensuring that the scheme was consistent, viz. all basis functions should add up to unity. The solution is advanced in a sequence of two half-steps, each of magnitude $\Delta t/2$ while the convolution, which forms the major part of the computational burden, is performed using steps of magnitude Δt. Figure 13 shows the basis functions associated with the first half-step and the second half-step, or so-called full-step.

The Half-step scheme requires significant modifications to the normal time marching algorithm (6). The new algorithm is given by

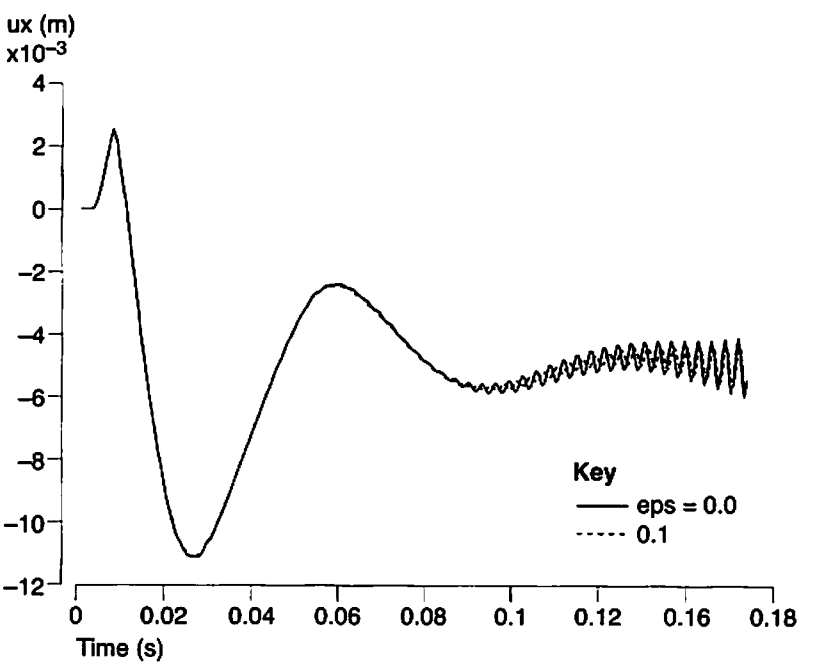


Figure 17.
Standard versus ε-scheme results for Hook problem

$$\underline{\underline{C}}^h_0 \underline{F}_{2m-1} + \sum_{k=0}^{m-1} \underline{\underline{C}}^h_{2m-1} \underline{F}_{2k} = \underline{b}_{2m-1}$$

$$\underline{\underline{C}}^f_0 \underline{F}_{2m} + \underline{\underline{C}}^f_1 \underline{F}_{2m-1} + \sum_{k=0}^{m-1} \underline{\underline{C}}^f_{2m-2k} \underline{F}_{2k} = \underline{b}_{2m} \quad (7)$$

where "h" and "f" imply half- and full-steps, respectively. Figure 13 depicts the shapes of the hat functions needed to calculate the influence coefficients at each time step for the new scheme.

The Trapezoidal time stepping algorithm (Figure 11) is summarized in the antiplane strain case (for simplicity only) in Appendix 1. The Half-step algorithm is summarized in Appendix 2 (please note that the authors make no claims as to the elegance of the code).

Time stepping scheme	Number of steps	Run time	Q1	Stable?
Trapezoidal	150	6.2 min	0.6	yes
	300	20.9 min	0.6	no
	75	2.1 min	1.2	no
	150	3.3 min	1.2	yes
	300	9.9 min	0.6	yes

Table II.
Run times for Trapezoidal versus Half step scheme

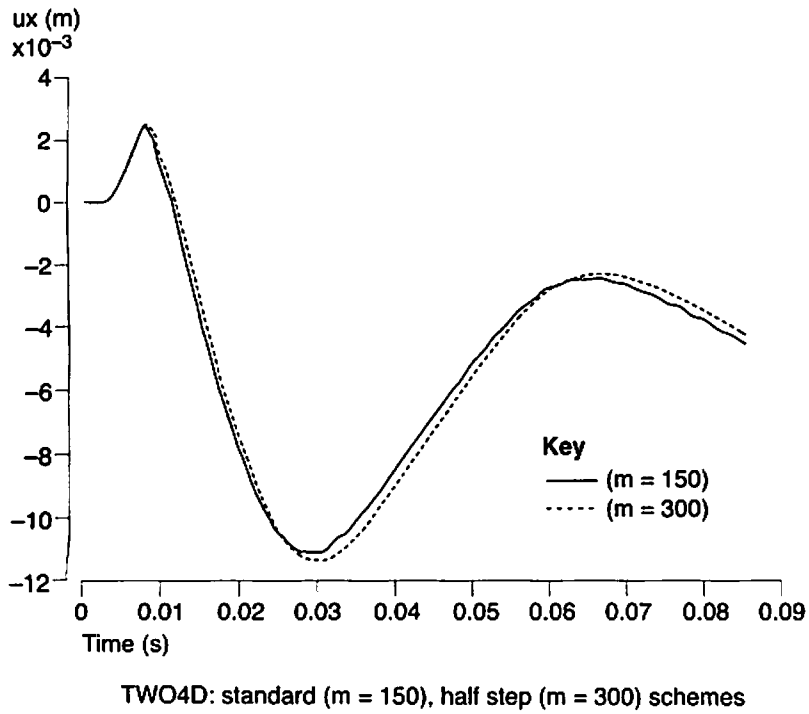


Figure 18.
Standard and Half-step results for Hook problem

Because the time step for each half-step in the new algorithm (7) is taken as half the time step used in the old algorithm (6), twice as many time steps are needed to advance the solution by the same amount of time. The number of coefficient matrices that are required in the new algorithm is twice that required by the old algorithm. However, the number of calculations that are required does not increase accordingly, because of the structure of the new algorithm. The largest share of CPU time is taken up by the calculation of the convolution histories at each time step, and this operation is only slightly slower in the new scheme than in the old scheme (even though there are twice the number of steps). Computer run times increase by about 50 per cent, but accuracy and stability are substantially improved. However, because of the increased accuracy of the new scheme, it is possible to use larger time steps. This implies that the new scheme is competitive with the old one (e.g. compare the run times of the two 150 time step runs in Table II). Table II and Figure 18 compare the two schemes in terms of CPU time and accuracy, respectively. Results are shown for the 28 element plane strain constant/linear Hook problem, run on a Pentium 66MHz machine. Figure 19 shows the two schemes plotted against results from *the* ABAQUS/EXPLICIT (HKS, 1994) finite element code. The Half-step scheme is clearly more accurate than the Trapezoidal scheme.

Figure 20 shows the Hook results obtained from the linear/linear plane strain version of TWO4D with $Q1 = 0.6$, and a numerical instability is evident by 500 time steps. A stability analysis predicts two poles(Peirce and Siebrits, 1995). Figure 21 shows the Half-step results to 3,000 time steps and there is no sign of an instability.

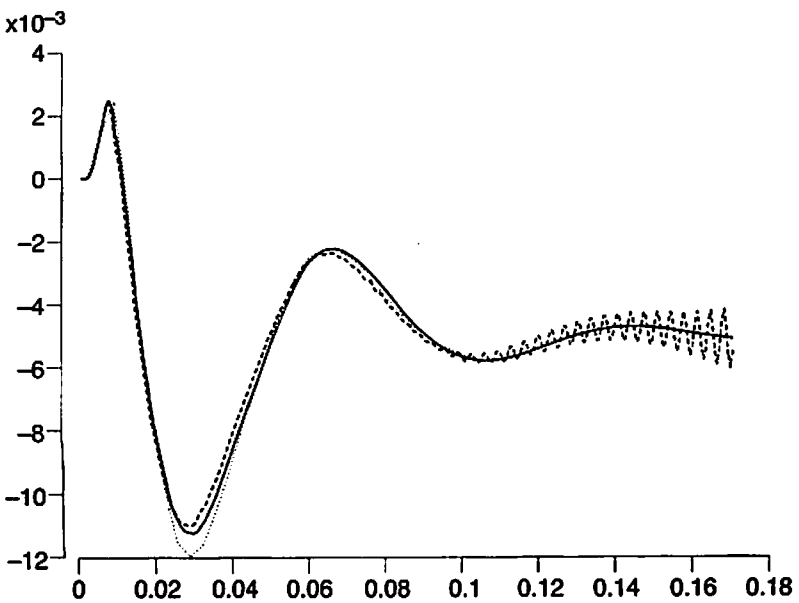
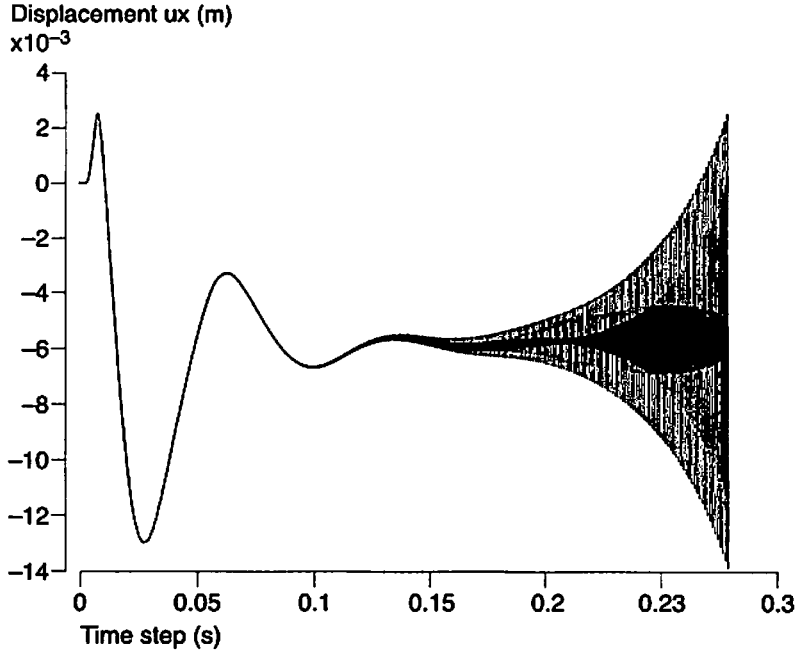
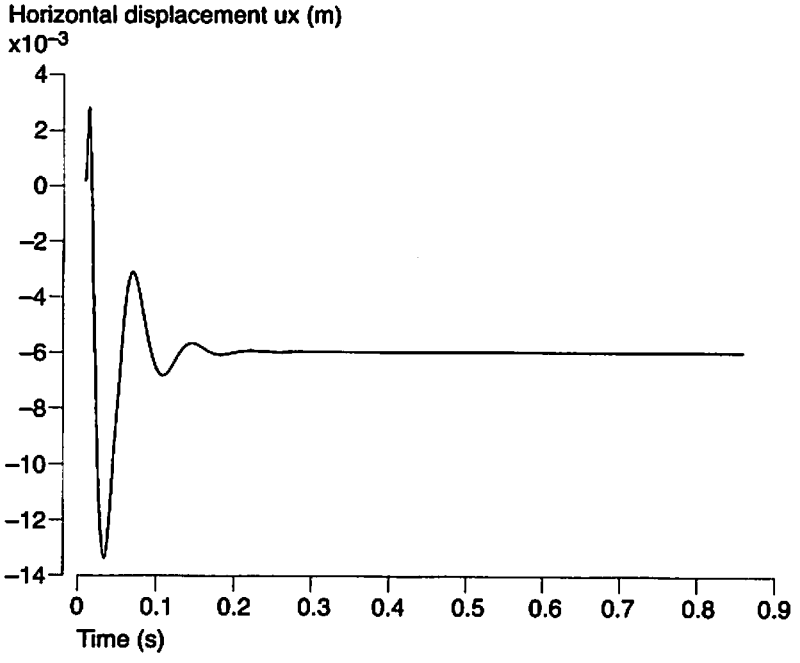


Figure 19.
Standard, Half-step, and ABAQUS/EXPLICIT results for Hook problem



TWO4Dv2: Hook, L/L, = 0.6, nt = 500

Figure 20.
Standard linear/linear Hook problem, showing numerical instability



TWO4D: Hook, L/L, half Q1 = 0.6, nt = 3000

Figure 21.
Half-step linear/linear Hook problem

The Half-step scheme was also implemented in the fictitious stress method, tested on the Selberg problem, and was found to show no improvement in the accuracy or stability properties of this method over the Trapezoidal time stepping scheme. The reason for this becomes clear if we consider that the response of the constant in space fictitious stress element to a triangular time impulse is a diffracted perturbation to the input pulse itself (Peirce and Siebrits, 1995). In other words, the stress self-effect kernels in the fictitious stress method are of order $O(1)$. However, in the displacement discontinuity method, the stress self-effects are of order $O(1/\Delta t)$ and the response is thus a sequence of step pulses whose magnitudes are determined by the gradients of the triangular input pulses (Peirce and Siebrits, 1995). Because the leading pulse of the Half-step scheme has double the gradient of the subsequent pulses (due to a time step half that of the Trapezoidal scheme), when it is applied to the displacement discontinuity method, the self effect is double that of the subsequent response pulses. This results in enhanced stability characteristics. However, when the Half-step scheme is applied to the fictitious stress method, the magnitude of the self effect is the same as it is for the Trapezoidal scheme. Therefore the Half-step scheme will not enhance the stability of the fictitious stress method.

Implementation of the Half-step scheme in the direct boundary element method shows a significant improvement in the accuracy and stability properties of this method for traction boundary value problems but not for displacement boundary value problems (Birgisson *et al.*, 1997). This can be explained in the following way. Recall that, in the two-dimensional direct boundary element and fictitious stress methods the integrated stress and displacement self effect kernels are of the orders $O(1)$ and $O(\Delta t)$, respectively. In the two-dimensional displacement discontinuity method, the stress and displacement self effect kernels are of the orders $O(1/\Delta t)$ and $O(1)$, respectively. In light of the above discussion on the application of the Half-step scheme to the fictitious stress method, we can see that the use of the Half-step scheme in the direct boundary element method will only be beneficial for traction boundary value problems because, even though the magnitude of the self effect is not enhanced by the smaller time step, the displacement kernel is reduced. This has the same effect as increasing the traction self effect kernel because the displacement self effect kernel is divided by the traction kernel during solution. In a displacement boundary value problem, a more stable time stepping scheme could be one in which a double-step scheme is used, i.e. the opposite of the Half-step scheme!

Other schemes

There are numerous other schemes that have been attempted, some of which will only be briefly mentioned here, because none of them are found to be as satisfactory as the Half-step scheme. For example, the Half-step scheme can be extended into a quarter-step scheme. Such a scheme has an even larger self effect than that for the Half-step scheme, which is beneficial for stability. However, it is found that the quarter-step scheme does not eliminate all numerical instabilities in TWO4D, besides requiring more time steps, storage

and being even more complicated to implement than the Half-step scheme, and is therefore rejected.

Various adaptations of the self effect can be postulated. For example, the first time ramp at the first time step can be changed into a quadratic ramp (Peirce and Siebrits, 1995). The benefit of such a time variation is that the self effect is larger, which is better for stability. However, this scheme follows an incorrect solution path because of an inconsistency between the leading quadratic influence and the trailing linear ones in the history terms, and is therefore rejected immediately. The use of quadratic Lagrange basis functions in a mixed quadratic and linear scheme results in a self effect that is smaller than some of the subsequent pulses. We can therefore expect such a scheme to be unstable.

Another time stepping scheme considered by the authors used Hermite cubic basis functions. Analysis shows (Peirce and Siebrits, 1995) that for this set of basis functions the self effect is very small, and this scheme can thus be discarded immediately. Analysis of a model problem shows that lower order time schemes are much better for stability (as opposed to accuracy) purposes than higher order ones (Peirce and Siebrits, 1995).

Conclusions

Amid the growing evidence of numerical instabilities in boundary element elastodynamic models, there has been little theoretical investigation of the causes of these instabilities and the possible strategies for remediation. We have provided evidence of the instabilities in a number of different formulations of the boundary element method.

To focus on the causes of these instabilities, we identified a simple example involving two parallel displacement discontinuity elements which rapidly exhibit numerical instabilities. We further expanded on the causes of instabilities with reference to the displacement discontinuity method.

We outlined the practical implementation of various time stepping schemes in boundary element algorithms. We discussed the implementation of schemes which assume that the unknown field is constant in time, which is suitable for the direct and fictitious stress indirect methods. We provided numerical results for the constant in time schemes for which different collocation points were used and we contrasted their stability.

We detailed the practical implementation of two new time stepping schemes for linear in time fields, namely the ε-scheme and the Half-step scheme. The enhanced stability characteristics of these new schemes were clearly demonstrated by comparing them with the standard or Trapezoidal time stepping schemes commonly used in boundary element algorithms. The enhanced stability characteristics of the new schemes were explained in terms of their modified influence characteristics.

Implementation of the Half-step scheme into the direct and indirect boundary element methods will allow boundary element users to attack far more challenging problems than before. We hope that researchers will find a complete cure to the problem of numerical instabilities in dynamic boundary element methods, so that these methods can be used to their full potential.

References

Andrews, D.J. (1994), "Dynamic growth of mixed-mode shear cracks, *Bull. Seism. Soc. Am.*, Vol. 84, pp. 1184-98.

Banerjee, P.K., Ahmad, S. and Manolis, G.D. (1987), "Advanced elastodynamic analysis", in Beskos (Ed.), *Boundary Element Methods in Mechanics*, pp. 258-84.

Birgisson, B., Peirce, A.P. and Siebrits, E. (1997), "Elastodynamic direct boundary element methods with enhanced numerical stability properties", *Int. J. Num. Meth. Engrg.*.

Dominguez, J. (1993), *Boundary Elements in Dynamics*, Computational Mechanics Publications, Southampton.

Eringen, A.C. and Suhubi, E.S. (1975), *Elastodynamics: Volume II Linear Theory*, Academic Press, New York, NY.

Graff, K.F. (1975), *Wave Motion in Elastic Solids*, Clarendon Press, Oxford.

Hibbitt, Karlsson, and Sorensen, Inc. (HKS) (1994), ABAQUS/EXPLICIT version 5.3-1, Rhode Island.

Kobayashi, S. (1985), "Fundamentals of boundary integral equation methods in elastodynamics", in Brebbia (Ed.), *Topics in Boundary Element Research, Vol. 2: Time-dependent and vibration problems*, Springer-Verlag, New York, NY, pp. 1-54.

Koller, M.G., Bonnet, B. and Madariaga, R. (1992), "Modelling of dynamical crack propagation using time-domain boundary integrals", *Wave Motion*, Vol. 16, pp. 339-66.

Loken, M.C. (1992), *A Three-Dimensional Boundary Element Method for Linear Elastodynamics*, Ph.D. Thesis, University of Minnesota, USA.

Love, A.E. (1944), *A Treatise on the Mathematical Theory of Elasticity*, Dover, New York, NY.

Mack, M.G. (1991), *A Three-Dimensional Boundary Element Method for Elastodynamics*, PhD. Thesis, University of Minnesota, USA.

Manolis, G.D., Ahmad, S. and Banerjee, P.K. (1986), "Boundary element method implementation for three-dimensional transient elastodynamics", in Banerjee and Watson (Eds), *Developments in Boundary Element Methods-4*, Elsevier, New York, NY, pp. 29-65.

Peirce, A.P. and Siebrits, E. (1995), *The Stability Properties of Time Domain Elastodynamic Boundary Element Methods*, Internal Note RE5/95, CSIR Mining Technology, Johannesburg, p. 108.

Selberg, H.L. (1951), "Transient compression waves from spherical and cylindrical cavities", *Arkiv for Fysik*, Vol. 5 No. 7, pp. 97-108.

Siebrits, E. (1992), *Two-Dimensional Time Domain Elastodynamic Displacement Discontinuity Method with Mining Applications*, PhD thesis, University of Minnesota, USA.

Tian, Y. (1990), *Boundary Element Method in Elastodynamics*, PhD thesis, University of Minnesota, USA.

Tikhonov, A.N. and Arsenin, Y.Y. (1977), *Solutions to Ill-posed Problems*, Winston-Wiley, New York, NY.

Appendix 1. Standard time stepping scheme

In the antiplane strain case, the standard time stepping scheme can be summarized as

```
MAIN
PARAMETER (NEQU = 100, NTIM = 1000)
DIMENSION C(NEQU, NEQU), P(NTIM, NEQU)
DO 100 K = 1, NTIM   ! time loop to precalculate influence coefficients
    DO 200 I = 1, NEQU   ! receiver loop
        DT = QI*DELX/CI   ! time step size
        DO 200 J = 1, NEQU   ! source loop
            SYZ3 = 0.0   ! initialize dummies
            SYZ2 = 0.0
```

```
                SYZ0 = 0.0
                CALL COEFF(I, J, K, SYZ)   ! 3 coefficient calls to build hat function
                SYZ3 = SXZ
                CALL COEFF(I, J, K-1, SYZ)
                SYZ2 = SYZ
                CALL COEFF(I, J, K-2, SYZ)
                SYZ0 = SYZ
                C(I,J) = SYZ3 - 2.0*SYZ2 + SYZ0   ! accumulate hat function
    200 CONTINUE
    WRITE(unit = 20) C   ! store influences for this time step
100 CONTINUE
REWIND(unit = 10)   ! rewind file
DO 400 M = 1, NTIM   ! step through all time
    READ(unit = 10)B   ! retrieve known boundary conditions from file
    IF(M .EQ. 1) GOTO 410   ! skip history loops if first time step
    REWIND(unit = 20)   ! rewind influences file
    READ(unit = 20)C   ! read first time step influences
    DO 500 K = 2, M   ! accumulate causal history effects
        READ(unit = 20)C   ! retrieve influences from file
        DO 600 I = 1, NEQU   ! receiver loop
            DO 600 J = 1, NEQU   ! source loop
                B(I) = B(I) - C(I, J)*P(M-K+1, J)   ! vector of knowns
        600 CONTINUE
    500 CONTINUE
    410 REWIND(unit = 20)
    READ(unit = 20)C   ! retrieve influence matrix for first time step
    CALL SOLVE(B, P, C)   ! solve equations
400 CONTINUE
WRITE(unit = 40)P   ! solution to file
END
```

Appendix 2. Half-step time stepping scheme

In the antiplane strain case, the Half-step algorithm can be summarized as (please note that the authors make no claims as to the elegance of the code, but it does work!)

```
MAIN
PARAMETER (NEQU = 100, NTIM = 2000)
DIMENSION C(NEQU, NEQU), P(NTIM, NEQU), CHALF(NEQU, NEQU)
DO 100 K = 1, NTIM   ! time loop to precalculate influence coefficients
    DO 200 I = 1, NEQU   ! receiver loop
        DT = QI*DELX/CI/2.0   ! time step size = half of full step size
        DO 200 J = 1, NEQU   ! source loop
            SYZ3 = 0.0   ! initialize dummies
            SYZ2 = 0.0
            SYZ0 = 0.0
            SYZ3A = 0.0
            SYZ2A = 0.0
            SYZ0A = 0.0
            TEST = AMOD(FLOAT(K), 2.0)   ! check for half (odd) or full (even) step
            IF(TEST .EQ. 0.0)THEN
                IF(K .EQ. 2)THEN
                CALL COEFF(I, J, K-1, SYZ)
                SYZ3A = SYZ
                CHALF(I, J) = SYZ3A
```

```
                    CALL COEFF(I, J, K, SYZ)
                    SYZ3 = SYZ
                    CALL COEFF(I, J, K-1, SYZ)
                    SYZ2 = SYZ
                    C(I, J) = SYZ3 - 2.0*SYZ2
                ELSE IF(K .EQ. 4)THEN
                    CALL COEFF(I, J, K, SYZ)
                    SYZ3 = SYZ
                    CALL COEFF(I, J, K-2, SYZ)
                    SYZ2 = SYZ
                    CALL COEFF(I, J, K-3, SYZ)
                    SYZ0 = SYZ
                    C(I, J) = 0.5*SYZ3 - 1.5*SYZ2 + SYZ0
                ELSE
                    CALL COEFF(I, J, K, SYZ)
                    SYZ3 = SYZ
                    CALL COEFF(I, J, K-2, SYZ)
                    SYZ2 = SYZ
                    CALL COEFF(I, J, K-4, SYZ)
                    SYZ0 = SYZ
                    C(I, J) = 0.5*SYZ3 - SYZ2 + 0.5*SYZ0
                ENDIF
            ELSE
                IF(K .EQ. 1)THEN
                    CALL COEFF(I, J, K, SYZ)
                    SYZ3A = SYZ
                    CHALF(I, J) = SYZ3A
                ELSE IF(K .EQ. 3)THEN
                    CALL COEFF(I, J, K, SYZ)
                    SYZ3A = SYZ
                    CALL COEFF(I, J, K-2, SYZ)
                    SYZ2A = SYZ
                    CHALF(I, J) = 0.5*SYZ3A - 1.5*SYZ2A
                ELSE
                    CALL COEFF(I, J, K, SYZ)
                    SYZ3A = SYZ
                    CALL COEFF(I, J, K-2, SYZ)
                    SYZ2A = SYZ
                    CALL COEFF(I, J, K-4, SYZ)
                    SYZ0A = SYZ
                    CHALF(I, J) = 0.5*SYZ3A - SYZ2A + 0.5*SYZ0A
                ENDIF
            ENDIF
        200 CONTINUE
        IF(TEST .EQ. 0.0)THEN   ! store appropriate odd and even influences
            IF(K .EQ. 2)WRITE(unit = 20)CHALF
            WRITE(unit = 20)C
        ELSE
            WRITE(unit = 30)CHALF
        ENDIF
    100 CONTINUE
    REWIND(unit = 10)   ! rewind boundary conditions file
    DO 400 M = 1, NTIM   ! step through all time
        TEST = AMOD(FLOAT(M), 2.0)
```

```
      READ(unit = 10)B   ! retrieve known boundary conditions from file
      IF(M .EQ. 1) GOTO 410   ! skip history loops if first time step
      REWIND(unit = 20)   ! rewind files
      REWIND(unit = 30)
      READ(unit = 20)C   ! read first time steps
      READ(unit = 30)CHALF
      MDUM = M
      IF(TEST .NE. 0.0)MDUM = M-1
            DO 500 K = 2, MDUM, 2   ! accumulate history effects in DOUBLE steps
                  IF(TEST .EQ. 0.0)THEN
                        READ(unit = 20)C
                  ELSE
                        READ(unit = 30)CHALF
                  ENDIF
                  DO 600 I = 1, NEQU   ! receiver loop
                        DO 600 J = 1, NEQU   ! source loop
                              IF(TEST .EQ. 0.0)THEN   ! build vector of knowns
                                    IF(K .EQ. 2)THEN
                                      B(I) = B(I) - C(I, J)*P(M-K+1, J)
                                    ELSE
                                      B(I) = B(I) - C(I, J)*P(M-K+2, J)
                                    ENDIF
                              ELSE
                                    B(I) = B(I) - CHALF(I, J)*P(M-K+1, J)
                              ENDIF
                  600 CONTINUE
            500 CONTINUE
      410 IF(TEST .EQ. 0.0)THEN   ! retrieve influences for first step
      REWIND(unit = 20)
      READ(unit = 20)C
      ELSE
            REWIND(unit = 30)
            READ(unit = 30)C
      ENDIF
      CALL SOLVE(B, P, C)   ! solve equations
400 CONTINUE
WRITE(unit = 40)P   ! solution to file
END
```

Conference calendar

24-28 November 1997
Kyoto, Japan
7th International Conference on Structural Safety and Reliability (ICOSSAR)
ICOSSAR Secretariat, c/o School of Civil Engineering, Kyoto University, Kyoto, 606-01, Japan. Tel: 81-75-753-5093; Fax: 81-75-761-0646; E-mail: icos97@brdgeng.gee.kyoto-u.ac.jp

January 1998
University of Arizona, Tucson, Arizona, USA
FLOW '98 – 10th International Conference on Finite Elements in Fluids
Professor R. Gallagher. Tel: (315) 268 6444; E-mail: dick5762@aol.com

11-13 February 1998
University of Sydney, Australia
ACSO'98: Australian Conference on Structural Optimisation
Dr Hong Guan. ACSO'98 Secretariat, Tel: +61 2 9351 2342 or 9351 2339, Fax: +61 2 9351 4841, E-mail: acso98@acro.ae.su.0z.au, Web: http://www.ae.su.oz.au/wwwdocs/ego1.html

31 March-3 April 1998
Bad Gastein, Austria
EURO-C 1998: Computational Modelling of Concrete Structures
Professor H. Mang, Institute for Strength of Materials, Vienna University of Technology, Karisplatz 13/202, A-1040 Vienna, Austria, Tel: +43 (1) 58801 3084; Fax: +43 (1) 504 1629; e-mail: herbert.mang@tuwien.ac.at

17-20 May 1998
San Diego Marriott La Jolla, California, USA
ASCE 12th Engineering Mechanics Division Conference
Professor J.E. Luco. Tel: (619) 534 4338; Fax: (619) 534 5698; E-mail: EM98_ASCE@ames.ucsd.edu

29 June-2 July 1998
Buenos Aires, Argentina
IACM – Fourth World Congress on Computational Mechanics
IACM Secretariat, Spain. Tel: (34) 3 401 6036; Fax: (34) 3 401 6517; E-mail: semni@etseccpb.upc.es

6-8 August 1998
Hyatt Regency in Embarcadero Centre, San Francisco, California, USA
The Fourth US National Congress on Computational Mechanics
Mark S. Shephard. Tel: (518) 276 6795; Fax: (518) 276 4886; E-mail: usnccm@scorec.rpi.edu

26-28 August 1998
Montreal, Canada
CADCOMP 98: Computer Methods in Composite Materials
Liz Kerr. Tel: +44 01703 293223; Fax: +44 01703 292853; e-mail: liz@wessex.ac.uk