Computer Methods for Nonlinear Solids and Structural Mechanics (eds. S.N. Atluri and N. Perrone), AMD-Vol. 54, ASME, New York (1983) 75-109.

NEW ALTERNATING DIRECTION PROCEDURES IN FINITE ELEMENT ANALYSIS BASED UPON EBE APPROXIMATE FACTORIZATIONS

T. J. R. Hughes, Professor and **J. Winget**, Graduate Research Assistant, California Institute of Technology
Division of Applied Mechanics
Department of Mechanical Engineering
Stanford University
Stanford, California

I. Levit, Assistant Professor, formerly graduate Research Assistant
California Institute of Technology and Stanford University
Faculty of Engineering
Department of Solid Mechanics, Materials and Structures
Tel Aviv University
Tel Aviv, Israel

T. E. Tezduyar, Assistant Professor, Formerly graduate Research Assistand
California Institute of Technology and Stanford University
Department of Mechanical Engineering
University of Houston
Houston, Texas

Abstract

Element-by-element approximate factorization procedures are proposed for solving the large finite element equation systems which arise in computational mechanics. A variety of techniques are compared on problems of structural mechanics, heat conduction and fluid mechanics. The results obtained suggest considerable potential for the methods described.

1. Introduction

Despite the attainment of significant increases in computer storage and speed in recent years, many contemporary problems of engineering interest are simply too complex to be solved with existing numerical algorithms and presently-available hardware. This is contrary to impressions created by the popular media that computer power is virtually infinite and abundantly available at small cost. The opposite reality is summarized by an often quoted pun, a variant of which asserts that there are two physical constants, c_1 and c_2, which characterize the storage capacity and speed, respectively, of all present and future computers. The values of these constants are c_1 = "too small" and c_2 = "too slow". Because it is anticipated that, for the forseeable future, the engineering appetite for computer power will far exceed its availability, it appears that the only recourse is the development of new algorithms which more fully exploit computational resources. In fact, it is interesting to note that Dean Chapman, in his Dryden Lecture of a few years ago [C1], compared the improvements made in hardware with computational aerodynamics algorithms over a fifteen year period and found that the improvement in algorithms equalled that for hardware. It is the opinion of the authors that there is still enormous potential for progress along these lines in computational fluid dynamics, as well as in structural mechanics and heat transfer analysis.

In this paper we address the subject of solving the matrix equations arising from finite element spatial discretizations. In Appendix I a brief sketch is given of how finite element equation systems emanate from various classes of continuum mechanical problems, such as typical structural, heat conduction, and fluids problems. The matrix equations, though sparsely populated, still entail

enormous storage demands, especially in three-dimensional cases. This is the major drawback to matrix-based ("implicit") finite element procedures. The types of methods we have developed to deal with this circumvent the need to form and factorize large global arrays. These methods have their origins in procedures which pervade the numerical analysis literature. Basically, the idea is to replace a large, complicated array by a product of simpler arrays. The original concepts apparently emanate from the so-called "alternating direction (ADI) methods" of Douglas [D3, D4], Douglas and Rachford [D7] and Peaceman and Rachford [P3]. There is a large Russian literature on methods of this type which is summarized in the books of Marchuk [M1] and Yanenko [Y1]. In these works the terminologies used are the "method of fractional steps", the "splitting-up method", and the "method of weak approximation". In the field of computational aerodynamics these techniques are often described as "approximate factorization" methods (see e.g. Warming and Beam [W1]). The preceding references deal primarily with finite difference methods in which the splitting is usually performed by decomposing a multi-dimensional partial differential operator into one-dimensional operators. This, of course, places geometrical and topological limitations on the discretizations. Generally these methods are used most effectively in the context of rectangular domains, or domains which are at least topologically equivalent to rectangles. When circumstances like this prevail, very large problems can be efficiently solved. As a case in point, we may mention the work of Rogallo [R1], in which an unsteady, three-dimensional Navier-Stokes simulation is performed on a grid of 128^3 points using an implicit, approximate factorization algorithm. A calculation of this magnitude would be inconceivable using standard implicit finite element algorithms.

Progress has been made in developing analogous finite element procedures (see Baker [B1], Douglas and Dupont [D5, D6], Dendy and Fairweather [D1], and Hayes [H1-H4]). However, these procedures do not retain the full geometric and topological versatility of finite element discretizations.

Glowinski and his colleagues have used a somewhat different approach to reduce the size of finite element systems (see e.g. [G1, G2]). The basic idea is to use a "subdomain modelling" philosophy similar in intent to the classical technique of "substructures". However, they employ iterative solvers, such as preconditioned conjugate gradients, to solve the global system. A number of impressive large-scale fluid dynamical computations have been performed in this way. This approach may also be viewed as a type of ADI, or approximate factorization, procedure. Here the simpler arrays are the subdomain models, which appears very natural in the finite element context.

The methods advocated herein have much in common with Glowinski's and derive many other features from the preceding references. The approximate factorization aspect of the present approach is facilitated by what we feel are the most simple and natural constituents of the finite element process--the individual element arrays. No more than one element array needs to be formed and stored at one time and calculations proceed in an element-by-element (EBE) fashion. There is no geometric or topological restriction imposed by the method, and at the same time a remarkably concise computational architecture is achieved. It is pointed out herein that the present approach has significant advantages when implicit-explicit finite element mesh partitions are employed, and, what appears to be most significant for the future, the method is amenable to parallel calculations on multi-processor computers.

The idea of element-by-element factorizations was first proposed in Hughes, Levit and Winget [H12] in which a transient algorithm for heat conduction was developed. Based upon this work, Ortiz, Pinsky and Taylor [O1] constructed a novel time-stepping scheme for dynamics. However, our research revealed stringent accuracy requirements in certain circumstances, and we were led to reformulate the procedure as an iterative linear equation solver (see Hughes, Levit and Winget [H13]). In this way the usual accuracy and stability properties of standard finite element algorithms is attained. The problems that we have applied these procedures to are all time dependent and mostly nonlinear. Nour-Omid and Parlett [N2] have applied similar procedures to static structures problems

and also report encouraging results.

An outline of the remainder of the paper follows: In Section 2 we describe two candidate iterative algorithms which can be used in conjunction with approximately-factorized arrays. In Section 3 various types of approximate factorizations are described including several EBE factorizations. In Section 4 the form of preconditioning matrix, which is ultimately approximately factorized, is delineated. In Section 5 we present some sample problems in structural mechanics, heat conduction and fluid dynamics. Comparisons are made between the candidate techniques. Conclusions are drawn in Section 6.

2. Iterative Algorithms

A variety of algorithms may be employed in conjunction with approximately-factorized arrays. The following two have been used in the numerical work performed by us so far.

2a. Parabolic Regularization (Hughes, Levit and Winget [H13])

The derivation of this algorithm is based upon replacing the algebraic problem

$$\underset{\sim}{A}\ \underset{\sim}{x} = \underset{\sim}{b}$$

by a first-order ordinary differential equation whose asymptotic solution is $\underset{\sim}{x}$. The terminology "parabolic regularization" is used since the algebraic problem is replaced by what amounts to a spatially-discrete parabolic problem. The ordinary differential equation is discretized by backward differences and the implicit operator is approximately factorized. Quasi-Newton updates and line searches are employed to accelerate convergence. The flowchart in Table 1 summarizes the procedure for symmetric positive-definite arrays. Further details may be found in [H13].

Table 1 Flowchart of the parabolic regularization (PR) algorithm with line search and BFGS updates

Step 1. Initialization:

$$m = 0 \quad , \quad \underset{\sim}{x}_0 = \underset{\sim}{0} \quad , \quad \underset{\sim}{r}_0 = \underset{\sim}{b}$$

$$\underset{\sim}{f}_k = \underset{\sim}{g}_k = \underset{\sim}{0} \qquad (\text{loop: } k = 1\ ,\ 2\ ,\ \ldots,\ n_{BFGS})$$

$$\Delta\underset{\sim}{x} = \underset{\sim}{B}^{-1}\ \underset{\sim}{r}_0$$

Step 2. Line search:

$$s = \Delta\underset{\sim}{x}^T\ \underset{\sim}{r}_m / \Delta\underset{\sim}{x}^T\ \underset{\sim}{A}\ \Delta\underset{\sim}{x}$$

$$\underset{\sim}{x}_{m+1} = \underset{\sim}{x}_m + s\ \Delta\underset{\sim}{x}$$

Step 3. Convergence check:

$$||\underset{\sim}{r}_{m+1}|| < \delta$$

Yes: Return

No : Continue

Step 4. Relabel old BFGS vectors:

$$\underset{\sim}{f}_{k-1} = \underset{\sim}{f}_k \quad , \quad \underset{\sim}{g}_{k-1} = \underset{\sim}{g}_k \quad , \quad \text{(loop: } k = 2 \text{ , } 3 \text{ ,..., } n_{BFGS})$$

Step 5. Calculate new BFGS vectors:

$$\underset{\sim}{f}_{n_{BFGS}} = (\Delta\underset{\sim}{x}^T \underset{\sim}{r}_m)^{-1} \Delta\underset{\sim}{x}$$

$$\underset{\sim}{g}_{n_{BFGS}} = \underset{\sim}{r}_{m+1} - (1 - s^{\frac{1}{2}})\underset{\sim}{r}_m$$

Step 6. New search direction:

$$\underset{\sim}{z} = \underset{\sim}{r}_{m+1}$$

$$\underset{\sim}{z} \leftarrow \underset{\sim}{z} + (\underset{\sim}{f}_k^T \underset{\sim}{z})\underset{\sim}{g}_k \quad \text{(loop: } k = n_{BFGS} \text{ , } n_{BFGS} - 1 \text{ ,..., } 1)$$

$$\underset{\sim}{z} \leftarrow \underset{\sim}{B}^{-1} \underset{\sim}{z}$$

$$\underset{\sim}{z} \leftarrow \underset{\sim}{z} + (\underset{\sim}{g}_k^T \underset{\sim}{z})\underset{\sim}{f}_k \quad \text{(loop: } k = 1 \text{ , } 2 \text{ ,..., } n_{BFGS})$$

$$\Delta\underset{\sim}{x} = \underset{\sim}{z}$$

Step 7. $m \leftarrow m + 1$, go to Step 2.

The notation in Table 1 is given as follows: m is the iteration counter; the $\underset{\sim}{f}_k$'s and $\underset{\sim}{g}_k$'s are the BFGS vectors; n_{BFGS} is the maximum number of BFGS vectors allowed; $\underset{\sim}{B}$ is a matrix which approximates $\underset{\sim}{A}$, but is more easily factorized; s is the search parameter; $\underset{\sim}{x}_m$ is the m^{th} approximation of $\underset{\sim}{x}$; $\underset{\sim}{r}_m = \underset{\sim}{b} - \underset{\sim}{A}\underset{\sim}{x}_m$ is the corresponding residual; $||\underset{\sim}{r}_m||$ is its Eucilean length; and δ is a preassigned error tolerance.

Remark

The search parameter in step 2 is determined by minimizing the potential energy

$$P(s) = -(\underset{\sim}{x}_m + s\,\Delta\underset{\sim}{x})^T (\underset{\sim}{b} - \frac{1}{2}\underset{\sim}{A}(\underset{\sim}{x}_m + s\,\Delta\underset{\sim}{x})) \qquad (2.1)$$

While this criterion seems appropriate for symmetric positive-definite $\underset{\sim}{A}$, an alternative is needed for unsymmetric and/or indefinite matrices. In these cases we have selected s so as to minimize the length of the residual $||\underset{\sim}{r}_{m+1}||$. This leads to

$$s = (\underset{\sim}{A}\,\Delta\underset{\sim}{x})^T \underset{\sim}{r}_m / ||\underset{\sim}{A}\,\Delta\underset{\sim}{x}||^2 \qquad (2.2)$$

Furthermore, other updates, such as Broyden's [D2], may be more appropriate for the unsymmetric case.

2b. Preconditioned Conjugate Gradients

This algorithm is a generalization of the classical conjugate gradients method (see Hestenes-Stiefel [H5]) in which a "preconditioning" is performed using $\underset{\sim}{B}$, the matrix approximating $\underset{\sim}{A}$. The algorithm is summarized in Table 2.

Table 2 Flowchart of preconditioned conjugate gradients (CG)

Step 1. Initialization:

$m = 0 \quad , \quad \underset{\sim}{x}_0 = \underset{\sim}{0}$

$\underset{\sim}{r}_0 = \underset{\sim}{b}$

$p_0 = \underset{\sim}{z}_0 = \underset{\sim}{B}^{-1} \underset{\sim}{r}_0$

Step 2. $\alpha_m = \underset{\sim}{r}_m^T \underset{\sim}{z}_m / \underset{\sim}{p}_m^T \underset{\sim}{A} \underset{\sim}{p}_m$

Step 3. $\underset{\sim}{x}_{m+1} = \underset{\sim}{x}_m + \alpha_m \underset{\sim}{p}_m$

Step 4. $\underset{\sim}{r}_{m+1} = \underset{\sim}{r}_m - \alpha_m \underset{\sim}{A} \underset{\sim}{p}_m$

Step 5. Convergence check:

$||\underset{\sim}{r}_{m+1}|| < \delta$?

Yes: Return

No : Continue

Step 6. $\underset{\sim}{z}_{m+1} = \underset{\sim}{B}^{-1} \underset{\sim}{r}_{m+1}$

Step 7. $\beta_m = \underset{\sim}{r}_{m+1}^T \underset{\sim}{z}_{m+1} / \underset{\sim}{r}_m^T \underset{\sim}{z}_m$

Step 8. $\underset{\sim}{p}_{m+1} = \underset{\sim}{z}_{m+1} + \beta_m \underset{\sim}{p}_m$

Step 9. $m \leftarrow m + 1$, go to Step 2.

Remark 1. Glowinski et al. [G1, G2] (see also references therein) have successfully used the preconditioned conjugate gradients algorithm in their finite element work. The matrix which they employ as preconditioner is determined by way of various "incomplete Cholesky factorizations" (see e.g. Thomasset [T2] and references therein).

Remark 2. The CG method is not designed for unsymmetric $\underset{\sim}{A}$. However, the algorithm can always be applied to the normal equations $\underset{\sim}{A}^T \underset{\sim}{A} \underset{\sim}{x} = \underset{\sim}{A}^T \underset{\sim}{b}$, but this strategy may be ill-advised (see Paige and Saunders [P1]).

Remark 3. A fixed number of vectors is all that is needed in the CG method. This makes it computationally more attractive than the PR algorithm with BFGS updates, because a considerable number of BFGS vectors typically need to be stored.

3. Approximate Factorization

The convergence rate of the algorithms presented in the preceding section depend heavily upon the approximating matrix $\underset{\sim}{B}$. It may be noted that if $\underset{\sim}{B} = \underset{\sim}{A}$ then both algorithms immediately obtain the exact solution $\underset{\sim}{x}$. Numerous choices for $\underset{\sim}{B}$ are possible. To explore some of the possibilities we shall introduce the following notational scheme. Let

$$\underset{\sim}{A} = \underset{\sim}{L}_p(\underset{\sim}{A})\underset{\sim}{D}_p(\underset{\sim}{A})\underset{\sim}{U}_p(\underset{\sim}{A}) \qquad \text{(product decomposition)} \tag{3.1}$$

$$\underset{\sim}{A} = \underset{\sim}{L}_s(\underset{\sim}{A}) + \underset{\sim}{D}_s(\underset{\sim}{A}) + \underset{\sim}{U}_s(\underset{\sim}{A}) \qquad \text{(sum decomposition)} \tag{3.2}$$

where the subscripts p and s indicate "product" and "sum", respectively.

Equation (3.1) represents the Crout factorization. Thus $\underset{\sim}{L}_p$ and $\underset{\sim}{U}_p$ are lower and upper triangular matrices, respectively, with diagonal entries equal to 1, and $\underset{\sim}{D}_p(\underset{\sim}{A})$ is a diagonal matrix. If $\underset{\sim}{A}$ is symmetric, then $\underset{\sim}{L}_p(\underset{\sim}{A}) = \underset{\sim}{U}_p^T(\underset{\sim}{A})$. If the entries of $\underset{\sim}{D}_p$ are nonnegative, then we can write

$$\underset{\sim}{A} = \tilde{\underset{\sim}{L}}_p(\underset{\sim}{A})\tilde{\underset{\sim}{U}}_p(\underset{\sim}{A}) \tag{3.3}$$

where

$$\tilde{\underset{\sim}{L}}_p = \underset{\sim}{L}_p\underset{\sim}{D}_p^{\frac{1}{2}} \tag{3.4}$$

$$\tilde{\underset{\sim}{U}}_p = \underset{\sim}{D}_p^{\frac{1}{2}}\underset{\sim}{U}_p \tag{3.5}$$

When $\underset{\sim}{A}$ is symmetric positive-definite, (3.3)-(3.5) defines the Cholesky, or square-root, factorization.

In equation (3.2), $\underset{\sim}{L}_s$ and $\underset{\sim}{U}_s$ are lower and upper triangular matrices with diagonal entries equal to 0, and $\underset{\sim}{D}_s$ is diagonal. In analogy with the product decomposition, we may write

$$\underset{\sim}{A} = \tilde{\underset{\sim}{L}}_s(\underset{\sim}{A}) + \tilde{\underset{\sim}{U}}_s(\underset{\sim}{A}) \tag{3.6}$$

where

$$\tilde{\underset{\sim}{L}}_s = \underset{\sim}{L}_s + \frac{1}{2}\underset{\sim}{D}_s \tag{3.7}$$

$$\tilde{\underset{\sim}{U}}_s = \underset{\sim}{U}_s + \frac{1}{2}\underset{\sim}{D}_s \tag{3.8}$$

If $\underset{\sim}{A}$ is symmetric, then $\underset{\sim}{L}_s(\underset{\sim}{A}) = \underset{\sim}{U}_s(\underset{\sim}{A})^T$ and $\tilde{\underset{\sim}{L}}_s(\underset{\sim}{A}) = \tilde{\underset{\sim}{U}}_s(\underset{\sim}{A})^T$.

Remark 1. The decomposition (3.6)-(3.8) has figured in the transient analysis algorithms developed by Trujillo [T4, T5] and subsequently discussed by Park [P2].

Remark 2. Note that the net total storage required for the sum decomposition is exactly the same as for the original matrix. However, the product decomposition entails increased storage due to "fill-in" of zeros within the skyline. This is perhaps the major drawback of direct solution schemes such as Crout elimination.

Remark 3. If we ignore the line search and quasi-Newton update ingredients of the PR algorithm, then classical iterative algorithms are obtained by choosing $\underset{\sim}{B}$ as follows:

$$\underset{\sim}{B} = \underset{\sim}{D}_s(\underset{\sim}{A}) \qquad \text{(Jacobi method)} \tag{3.9}$$

$$\underset{\sim}{B} = \underset{\sim}{L}_s(\underset{\sim}{A}) + \underset{\sim}{D}_s(\underset{\sim}{A}) \qquad \text{(Gauss-Seidel method)} \tag{3.10}$$

To describe the procedures that are emphasized herein, we first consider

matrices, $\tilde{\underset{\sim}{A}}$, written in the following form:

$$\tilde{\underset{\sim}{A}} = \underset{\sim}{W}^{\frac{1}{2}}(\underset{\sim}{I} + \varepsilon\, \bar{\underset{\sim}{A}})\underset{\sim}{W}^{\frac{1}{2}} \tag{3.11}$$

where $\underset{\sim}{I}$ is the identity matrix, $\underset{\sim}{W}$ is a positive-definite diagonal matrix, ε is a scalar, and $\bar{\underset{\sim}{A}}$ is a matrix which has the same sparsity pattern as $\underset{\sim}{A}$. $\tilde{\underset{\sim}{A}}$ is to be thought of as an approximation of $\underset{\sim}{A}$. Specific choices of $\underset{\sim}{W}$, ε and $\bar{\underset{\sim}{A}}$ are considered later in section 4. The second and final stage of the approximation is to define

$$\underset{\sim}{B} = \underset{\sim}{W}^{\frac{1}{2}}\, \underset{\sim}{C}\, \underset{\sim}{W}^{\frac{1}{2}} \tag{3.12}$$

where $\underset{\sim}{C}$ is an approximation of $\underset{\sim}{I} + \varepsilon\, \bar{\underset{\sim}{A}}$. Various choices are considered below:

3a. Two-component Splitting

Let $\bar{\underset{\sim}{A}}$ be decomposed as follows:

$$\bar{\underset{\sim}{A}} = \bar{\underset{\sim}{A}}_1 + \bar{\underset{\sim}{A}}_2 \tag{3.13}$$

Then a possible definition of $\underset{\sim}{C}$ is

$$\underset{\sim}{C} = (\underset{\sim}{I} + \varepsilon\, \bar{\underset{\sim}{A}}_1)(\underset{\sim}{I} + \varepsilon\, \bar{\underset{\sim}{A}}_2) = \underset{\sim}{I} + \varepsilon\, \bar{\underset{\sim}{A}} + \varepsilon^2\, \bar{\underset{\sim}{A}}_1\bar{\underset{\sim}{A}}_2 = \underset{\sim}{I} + \varepsilon\, \bar{\underset{\sim}{A}} + O(\varepsilon^2) \tag{3.14}$$

The last line suggests the nature of the approximation. Computational simplicity is gained if $\bar{\underset{\sim}{A}}_1$ and $\bar{\underset{\sim}{A}}_2$ are very sparse and more easily factorized than $\underset{\sim}{A}$.

For example, let

$$\bar{\underset{\sim}{A}}_1 = \tilde{\underset{\sim}{L}}_S(\bar{\underset{\sim}{A}}) \tag{3.15}$$

$$\bar{\underset{\sim}{A}}_2 = \tilde{\underset{\sim}{U}}_S(\bar{\underset{\sim}{A}}) \tag{3.16}$$

Thus $\underset{\sim}{B}$ has the following simple form

$$\underset{\sim}{B} = \underset{\sim}{W}^{\frac{1}{2}}(\underset{\sim}{I} + \varepsilon\, \tilde{\underset{\sim}{L}}_S(\bar{\underset{\sim}{A}}))\ (\underset{\sim}{I} + \varepsilon\, \tilde{\underset{\sim}{U}}_S(\bar{\underset{\sim}{A}}))\underset{\sim}{W}^{\frac{1}{2}} \tag{3.17}$$

As may be seen, $\underset{\sim}{B}$ is already factored and the factors require no more storage than that for $\underset{\sim}{A}$. Only diagonal scaling and forward reductions and back substitutions with sparse triangular arrays are needed to solve equations with $\underset{\sim}{B}$ as coefficient matrix. This eliminates the cost of factorization and obviates the storage penalties due to "fill-in". Equation (3.17) represents a symmetrized Gauss-Seidel type approximate factorization.

3b. One-pass Multi-component Splitting

Consider a multi-component sum decompositon of $\bar{\underset{\sim}{A}}$:

$$\bar{\underset{\sim}{A}} = \sum_{i=1}^{n} \bar{\underset{\sim}{A}}_i \tag{3.18}$$

Let

$$\underset{\sim}{C} = \prod_{i=1}^{n} (\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}_i)$$

$$= (\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}_1)(\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}_2) \ \ldots \ (\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}_n)$$

$$= \underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}} + O(\varepsilon^2) \tag{3.19}$$

Clearly, this is just a straightforward generalization of the two-component splitting.

3c. Two-pass Multi-component Splitting

This generalization of the preceding case has qualitative advantages under certain circumstances (Marchuk [M1]). Let

$$\underset{\sim}{C} = \prod_{i=1}^{n} (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_i) \prod_{i=n}^{1} (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_i)$$

$$= (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_1)(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_2) \ \ldots \ (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_n) \ \times$$

$$\times \ (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_n)(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_{n-1}) \ \ldots \ (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}_1)$$

$$= \underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}} + O(\varepsilon^2) \tag{3.20}$$

If each $\overline{\underset{\sim}{A}}_i$ is symmetric and positive semi-definite, then $\underset{\sim}{C}$ is symmetric and positive-definite.

3d. Element-by-element (EBE) Approximate Factorizations

The EBE approximate factorization is simply a multi-component splitting in which the components are the finite element arrays themselves. That is we assume

$$\overline{\underset{\sim}{A}} = \sum_{i=1}^{n_{e\ell}} \overline{\underset{\sim}{A}}^e \tag{3.21}$$

where $\overline{\underset{\sim}{A}}^e$ is the e^{th} element contribution to $\overline{\underset{\sim}{A}}$. Then $\underset{\sim}{C}$ may be defined by either the one-pass or two-pass formulae, viz.

$$\underset{\sim}{C} = \prod_{e=1}^{n_{e\ell}} (\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}^e) \tag{3.22}$$

$$\underset{\sim}{C} = \prod_{i=1}^{n_{e\ell}} (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e) \prod_{e=n_{e\ell}}^{1} (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e) \quad \text{("Marchuk EBE")} \tag{3.23}$$

Remark 1. We wish to use the term element in the generic sense of a "subdomain model", where an element could be an individual finite element or a subassembly of elements. Thus we allow limited assembly. Various equivalent terminologies have been used to define this concept, such as "substructures" and "superelements". Subdomain finite element models inherit the symmetry and definiteness properties of the global array. Consequently, the remark made

after (3.20) applies.

Remark 2. The element arrays in (3.22) and (3.23) need to be factorized into triangular form. This can be done exactly using product decompositions or approximately using sum decompositions as in section 3a, equations (3.15)-(3.17):

one-pass

Corresponding to (3.22) we have

$$\underset{\sim}{C} = \prod_{e=1}^{n_{e\ell}} \underset{\sim}{L}_p(\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}^e)\underset{\sim}{D}_p(\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}^e)\underset{\sim}{U}_p(\underset{\sim}{I} + \varepsilon \overline{\underset{\sim}{A}}^e) \qquad \text{(product)} \qquad (3.24)$$

or

$$\underset{\sim}{C} = \prod_{e=1}^{n_{e\ell}} (\underset{\sim}{I} + \varepsilon \tilde{\underset{\sim}{L}}_s(\overline{\underset{\sim}{A}}^e))\ (\underset{\sim}{I} + \varepsilon \tilde{\underset{\sim}{U}}_s(\overline{\underset{\sim}{A}}^e)) \qquad \text{(sum)} \qquad (3.25)$$

Note (3.24) is identical to (3.22) whereas (3.25) is an approximation of (3.22).

two-pass

Corresponding to (3.23) we have

$$\underset{\sim}{C} = \prod_{e=1}^{n_{e\ell}} \underset{\sim}{L}_p(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e)\underset{\sim}{D}_p(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e)\underset{\sim}{U}_p(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e) \times$$

$$\times \prod_{e=n_{e\ell}}^{1} \underset{\sim}{L}_p(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e)\underset{\sim}{D}_p(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e)\underset{\sim}{U}_p(\underset{\sim}{I} + \tfrac{\varepsilon}{2} \overline{\underset{\sim}{A}}^e) \quad \text{(product)} \qquad (3.26)$$

or

$$\underset{\sim}{C} = \prod_{e=1}^{n_{e\ell}} (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \tilde{\underset{\sim}{L}}_s(\overline{\underset{\sim}{A}}^e))\ (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \tilde{\underset{\sim}{U}}_s(\overline{\underset{\sim}{A}}^e)) \times$$

$$\times \prod_{e=n_{e\ell}}^{1} (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \tilde{\underset{\sim}{L}}_s(\overline{\underset{\sim}{A}}^e))\ (\underset{\sim}{I} + \tfrac{\varepsilon}{2} \tilde{\underset{\sim}{U}}_s(\overline{\underset{\sim}{A}}^e)) \qquad \text{(sum)} \qquad (3.27)$$

Note (3.26) is identical to (3.23) whereas (3.27) is an approximation of (3.23).

Whether to use product or sum factorizations of the element arrays is a question of efficiency. Belytschko and Liu [B2] have proposed a fast, exact inversion procedure for 4-node heat conduction elements. For subassemblies, the approximate sum factorizations may have advantages.

Remark 3. Note that storage demands are vastly less in the EBE case. Only one element at a time need be stored and processed. Whether or not it is desirable to save factorized element arrays depends upon the availability of high speed

RAM, and the trade-off between CPU and disk I/O costs.

Remark 4. The ordering of the factors influences how well $\underset{\sim}{C}$ approximates $\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}}$. The global product decompostition,

$$\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}} = \underset{\sim}{L}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}})\underset{\sim}{D}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}})\underset{\sim}{U}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}}) \quad , \tag{3.28}$$

suggests that it might be worthwhile to reorder the factors in (3.24)-(3.27) such that all lower triangular factors precede diagonals which in turn precede upper triangular factors. This results in the following "reordered" schemes:

$$\underset{\sim}{C} = \left[\prod_{e=1}^{n_{e\ell}} \underset{\sim}{L}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}}^e)\right] \left[\prod_{e=1}^{n_{e\ell}} \underset{\sim}{D}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}}^e)\right] \times$$

$$\times \left[\prod_{e=n_{e\ell}}^{1} \underset{\sim}{U}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}}^e)\right] \qquad \text{("Crout EBE")} \tag{3.29}$$

$$\underset{\sim}{C} = \left[\prod_{e=1}^{n_{e\ell}} (\underset{\sim}{I} + \varepsilon \underset{\sim}{\tilde{L}}_s(\underset{\sim}{\bar{A}}^e))\right] \left[\prod_{e=n_{e\ell}}^{1} (\underset{\sim}{I} + \varepsilon \underset{\sim}{\tilde{U}}_s(\underset{\sim}{\bar{A}}^e))\right]$$

("symm. Gauss-Seidel EBE") (3.30)

Note that in the case of symmetric $\underset{\sim}{\bar{A}}$, symmetry is preserved by (3.29) and (3.30). Thus there seems little motivation for similarly reordering the two-pass versions.

In the case of positive $\underset{\sim}{D}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{A}^e)$'s, the Crout factorizations can be reordered in terms of Cholesky factors. For example, a variant of (3.29) is

$$\underset{\sim}{C} = \left[\prod_{e=1}^{n_{e\ell}} \underset{\sim}{\tilde{L}}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}}^e)\right] \left[\prod_{e=n_{e\ell}}^{1} \underset{\sim}{\tilde{U}}_p(\underset{\sim}{I} + \varepsilon \underset{\sim}{\bar{A}}^e)\right]$$

("Cholesky EBE") (3.31)

Note (3.31) and (3.29) are not generally identical.

Remark 5. If elements are segregated into non-contiguous subgroups then calculations are parallelizable. For example, brick-like domains can be decomposed into eight non-contiguous element groups (see Figure 1). Because the elements in each subgroup have no common degrees-of-freedom, they can be processed in parallel. The eight groups, however, need to be processed sequentially. For analogous two-dimensional domains, four element groups need to be employed.

Remark 6. It has been our computational experience that if $\underset{\sim}{A}$ is symmetric and positive-definite, then qualitatively faithful approximate factorizations, which preserve these properties, perform much better than those that do not. Consequently, in the numerical examples presented herein we have only employed qualitatively faithful approximate factorizations.

4. Selection of $\underset{\sim}{W}$, ε and $\overline{\underset{\sim}{A}}$.

The following three definitions of $\underset{\sim}{W}$, ε and $\overline{\underset{\sim}{A}}$ have been employed:

a.) This choice is motivated by the derivation of the PR algorithm (see [H13])

$$\underset{\sim}{W} = \underset{\sim}{D}_S(\underset{\sim}{A}) \tag{4.1}$$

$$\overline{\underset{\sim}{A}} = \frac{1}{\varepsilon} \underset{\sim}{W}^{-\frac{1}{2}} \underset{\sim}{A} \underset{\sim}{W}^{-\frac{1}{2}} \tag{4.2}$$

Thus

$$\tilde{\underset{\sim}{A}} = \underset{\sim}{D}_S(\underset{\sim}{A}) + \underset{\sim}{A} \tag{4.3}$$

b.) In this case

$$\underset{\sim}{W} = \underset{\sim}{D}_S(\underset{\sim}{A}) \tag{4.4}$$

$$\overline{\underset{\sim}{A}} = \frac{1}{\varepsilon} \underset{\sim}{W}^{-\frac{1}{2}} (\underset{\sim}{A} - \underset{\sim}{D}_S(\underset{\sim}{A})) \underset{\sim}{W}^{\frac{1}{2}} \tag{4.5}$$

which leads to

$$\tilde{\underset{\sim}{A}} = \underset{\sim}{A} \tag{4.6}$$

This procedure was proposed in Winget [W2].

Remark

Matrices of the form $\tilde{\underset{\sim}{A}} = \underset{\sim}{D}_S(\underset{\sim}{A}) + \varepsilon \underset{\sim}{A}$ were introduced in [H13]. Nour-Omid and Parlett [N2] analytically investigated the effectiveness of matrices of this type on a model problem and concluded that the optimal valve of ε was ∞. This limit is achieved by the definitions (4.4) and (4.5). ■

c.) For unsymmetric, indefinite cases we have employed

$$\underset{\sim}{W} = \text{diagonal of the mass matrix} \tag{4.7}$$

$$\overline{\underset{\sim}{A}} = \underset{\sim}{W}^{-\frac{1}{2}} \underset{\sim}{A} \underset{\sim}{W}^{-\frac{1}{2}} \tag{4.8}$$

which results in

$$\tilde{\underset{\sim}{A}} = \underset{\sim}{W} + \varepsilon \underset{\sim}{A} \tag{4.9}$$

The value of ε was picked on an experimental basis.

Remark

The implicit-explicit finite element concept [H10, H11, H14-H17] has a very simple and clean implementation within EBE approximate factorizations. Recall that an explicit element contributes only its diagonal mass matrix to the coefficient matrix $\underset{\sim}{A}$. Thus $\underset{\sim}{W}$, according to any one of the preceding definitions, totally accounts for the explicit element contributions and the corresponding $\overline{\underset{\sim}{A}}^e$'s are identically zero. What this means is that explicit elements may be simply omitted from the formula for $\underset{\sim}{C}$. In nonlinear problems this opens the way to time-adaptive implicit-explicit element partitions. In calculating the element contributions to the residual (i.e. "$\underset{\sim}{b}$") a check can be made whether or not the critical time step is exceeded for the element. If it is not exceeded, a flag is set to indicate that element contributions to $\underset{\sim}{C}$ may be simply ignored. The potential savings in nonlinear transient analysis procedures incorporating these ideas is clearly considerable.

5. Sample Problems

The computed results were obtained on a VAX computer using single precision (32 bits per floating point word).

5a. Structural Mechanics

The EBE calculations in this section were all performed with the PR algorithm of Table 1. No limit was set on the number of BFGS vectors (i.e., "$n_{BFGS} = \infty$," in Table 1). The selection of $\underset{\sim}{W}$, ε and $\bar{A}$ is according to (4.1) and (4.2) resulting in $\underset{\sim}{A} = D_s(\underset{\sim}{A}) + \underset{\sim}{A}$. The two-pass EBE splitting was employed, (3.23), with exact element factorization, (3.26).

Elastic Cantilever Beam

The configuration analyzed is shown in Figure 2. It represents one-half of a plane strain beam modelled with 32 bilinear quadrilateral elements. A lumped mass matrix was employed. The loading and boundary conditions are set in accord with an exact, static linear elasticity solution (see pp. 35-39, [T3]). However, here the problem is forced dynamically. The beam is assumed initially at rest and all loads are applied instantaneously at $t = 0+$. In formulating the problem, the Newmark algorithm is employed with $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ (see Appendix I). With these parameters, unconditional stability is attained and no algorithmic damping is introduced [G3, H7, H11].

The numerical solution is dominated by response in the fundamental mode. This is illustrated in Figure 3. At a time step of $\Delta t = 2.5 \times 10^{-4}$, an essentially exact solution is obtained. At a larger step of $\Delta t = 2.5 \times 10^{-3}$, a very crude approximation of the response is obtained. It is interesting to relate the sizes of these steps to the critical time step for explicit integration, $\Delta t_{crit} = h_{min}/c_D = h_{min}/\sqrt{(\lambda + 2\mu)/\rho} = 1.336 \times 10^{-5}$, and the approximate period of the fundamental node, $T_1 \cong .0122$ (see Table 3). As may be seen, both time steps are far outside the range of explicit integration. The larger time step resolves the fundamental mode with only 5 steps, and thus is larger than the maximum feasible for this problem.

In comparing the results of the various methods it is important to keep in mind that all methods give identical solutions.* Consequently, the primary basis of comparison in the number of iterations needed to attain the solution. It was found that the number of iterations per time step did not vary significantly from one time step to another for a given method and specific step size. Results for the first time step are presented in Table 4. The following observations may be made: In general the element-by-element results are superior to Jacobi. Use of line search and BFGS updates accelerate convergence. The best results are attained by the element-by-element procedure with line search and BFGS updates.

It is somewhat surprising that methods (v) and (vi) converge faster at the larger time step than at the smaller. At this point we have no explanation for this phenomenon.

*The convergence criterion, δ in step 3 of the flowchart, was taken to be $.01||\underset{\sim}{b}||$.

Table 3 Comparison of time steps used in calculations with characteristic time scales.

Δt	2.5×10^{-4}	2.5×10^{-3}
$\frac{\Delta t}{\Delta t_{crit}}$	18.71	187.1
$\frac{T_1}{\Delta t}$	48.9	4.89

Table 4 Number of iterations required for convergence for the problem illustrated in Figure 2.

Method \ Δt	2.5×10^{-4} ($= 18.71\ \Delta t_{crit}$)	2.5×10^{-3} ($= 187.1\ \Delta t_{crit}$)
(i) Jacobi	99	∞(1)
(ii) Jacobi + LS	38	75
(iii) Jacobi + LS + BFGS	15	21
(iv) EBE	14	16
(v) EBE + LS	9	6
(vi) EBE + LS + BFGS	5	4

Key: LS = line search
EBE = element-by-element (2-pass Marchuk type)

Note: (1) No convergence attained after 150 iterations.

Elastic and Elastic-Perfectly Plastic Cantilever Beam

The geometrical definition of this problem is identical to the previous one except that the entire beam is discretized by a 64 element mesh (the lower part of the beam was added to the mesh of Figure 2). The boundary conditions were changed to the following.

$$\left.\begin{aligned} u_1(0, x_2, t) &= u_2(0, 0, t) = 0 \\ t_2(L, x_2, t) &= Q\left(\frac{2t}{T}\right)\left(1 - \left(\frac{x_2}{c}\right)^2\right) \end{aligned}\right\} \quad x_2 \in [-c, +c] \quad , \quad t \in [0, T]$$

$$Q = 1{,}000 \quad , \quad T = 0.04 \quad , \quad L = 16 \quad , \quad c = 2$$

The boundary tractions are zero on the remaining boundary segments. The tensile uniaxial yield stress was taken to be 3,000. Small deformations were assumed and the elastic stiffness matrix was used on the left-hand side throughout. The radial-return algorithm [K2] was employed to integrate the elastic-plastic constitutive equation.

Figures 4 and 5 compare the elastic and plastic stress distributions at $t = .036$. A fully developed plastic hinge is present at the root of the beam in the plastic case. The elastic critical time step of this problem is $\Delta t_{crit} = 1.336 \times 10^{-5}$ and the time step used was $\Delta t = 2.5 \times 10^{-3} = 187.1 \ \Delta t_{crit}$. The EBE method with BFGS updates and line searches was employed. The average number of iterations for both the elastic and plastic calculations was 4.

Elastic and Elastic-Perfectly Plastic Cantilever Beam with a Circular Hole

The geometric definition of the problem is shown in Figure 6. The beam was discretized using a 500 element mesh. The following kinematic and stress boundary conditions were employed:

$$\left.\begin{aligned} u_1(0, x_2, t) &= u_2(0, 0, t) \\ t_2(L, x_2, t) &= Q\left(\frac{t}{T}\right)\left(1 - \left(\frac{x_2}{c}\right)^2\right) \end{aligned}\right\} \quad x_2 \in [-c, +c] \quad , \quad t \in [0, T]$$

$$Q = 250 \quad , \quad T = 0.09 \quad , \quad L = 28 \quad , \quad c = 4$$

Where not specified to be nonzero, the tractions are zero. The uniaxial yield stress was taken to be 1000. A critical time step of $\Delta t_{crit} = 5.41 \times 10^{-6}$ was calculated on the basis of the smallest element edge length. (The critical time step based upon the shortest distance between opposite element edges, which may be a more meaningful distance, is less than half this number.) Two time steps were employed in the calculations: $\Delta t = 10 \times \Delta t_{crit}$ and $\Delta t = 50 \times \Delta t_{crit}$. The Marchuk EBE algorithm with BFGS updates was also employed for this problem. Results for the smaller time step converged in 1 iteration, whereas for the latter, 7 iterations were required on average.

Figure 7 compares the elastic displacement time history at the tip of the beam to the plastic solution. Figures 8 and 9 show the stress distributions at time $t = 0.09$ for the elastic and plastic solutions. A fully developed plastic hinge is present at the end of the beam and a secondary plastic hinge has partially developed in the stress concentration zone (plastic regions are shown

dashed in Figure 9c).

We wish to remark upon the contour-line routine used to obtain the results presented in this section. The finite element analyzer calculates the stresses at the integration points. The data is then extrapolated to the nodal points by means of a weighted average of all the integration points in the interior domains of all elements connected to the nodal point. The weights are taken as the inverse of the distance between the integration point and the nodal point. This type of data smoothing ensures that the values obtained at the nodal points will be bounded by the data calculated at the integration points which is an essential property when plastic stresses are present and ensures continuity of stress contours between element domains. However, this method has some drawbacks. For example, data which is anti-symmetric about the neutral axis of the beam, such as σ_{11}, will result in linear distribution of contour lines in all elements having the center line as part of their boundary even in the case where these elements have a uniform stress distribution (part of a plastic hinge). Data with symmetry with respect to the neutral axis, such as the von Mises stress, does not suffer from this type of smoothing.

5b. Heat Conduction

The Δt_{crit}'s for heat conduction problems were computed from $\Delta t_{crit} = 2/\lambda_{max}$ where λ_{max} is the maximum element eigenvalue. Throughout, bilinear quadrilaterals were employed with 2×2 Gauss integration.

NASA Insulated Structure Test Problem

The problem description is illustrated in Figure 10. A number of comparisons of the various techniques proposed were made for this problem. In Table 5a the selection of $\tilde{\underset{\sim}{A}} = \underset{\sim}{A}$ [W2] is seen to converge faster than $\tilde{\underset{\sim}{A}} = \underset{\sim}{D}_S(\underset{\sim}{A}) + \underset{\sim}{A}$. In addition the steady-state residual potential energy, measured by $\log_{10}(-P_f)$, attains a smaller value when $\tilde{\underset{\sim}{A}} = \underset{\sim}{A}$. This and other calculations have indicated that $\tilde{\underset{\sim}{A}} = \underset{\sim}{A}$ is the superior choice. It is used in the comparisons shown in Figure 5b. The first observation which may be made here is that the CG algorithm is more effective than PR with line searches. The use of BFGS updates would doubtlessly improve upon the performances of PR, however, the increased data pool required to store the BFGS vectors is a significant disadvantage. Thus our current preference in symmetric positive-definite cases is the CG method. The EBE factorizations, ranked from best to worst, are: Crout, Cholesky, Marchuk, and symmetrized Gauss-Seidel. Nevertheless, it must be kept in mind that overall computational efficiency may alter this ordering. For example, although symmetrized Gauss-Seidel was the slowest to converge, it does not require element factorization, an advantage. A final point to observe is that convergence is typically slower during the larger time step sequences (i.e. steps 21-50) than for the smaller step sequences (i.e. steps 1-20). The reason for this appears to be that for the larger steps the solution approximates the steady state and thus the initial residual is fairly small which results in a more stringent convergence criterion. A more reasonable convergence criterion would no doubt result in faster termination for the larger steps than for the smaller. In fact, even the "non-converged" solutions possessed adequate accuracy from a practical standpoint.

Parallel/Sequential Test Problem

The problem description is given in Figure 11. The purpose of this problem is to compare convergence characteristics for "natural" element orderings, which necessitate sequential processing, with orderings that lend themselves to parallel computations. The comparisons were all performed with the CG algorithm, $\tilde{\underset{\sim}{A}} = \underset{\sim}{A}$ and the Cholesky EBE approximate factorization.

Over the thirty time steps the sequential ordering averaged 2.53 iterations per step to attain convergence, whereas the parallel ordering averaged 3.47 iterations. Despite the fact that the parallel ordering is slower, which might

be anticipated, the fact that it is reasonably fast is extremely encouraging. For the 256 element mesh shown a 64-processor computer could attain speeds 64 times faster than a single processor. This more than compensates for the somewhat slower convergence of the parallel ordering. The gains in larger problems are potentially even more spectacular.

Table 5a. Comparison of $\tilde{\underset{\sim}{A}} = D_s(\underset{\sim}{A}) + \underset{\sim}{A}$ with $\tilde{\underset{\sim}{A}} = \underset{\sim}{A}$.

Algorithm: PR + LS (no BFGS)

Approximate factorization: Marchuk EBE

$\tilde{\underset{\sim}{A}}$	$\log_{10}(-P_f)^{(\dagger)}$	ave. it's. per step	
		steps 1-20	steps 21-50(‡)
$D_s(\underset{\sim}{A}) + \underset{\sim}{A}$	- 13.0	6.0	10
$\underset{\sim}{A}$	- 15.4	5.03	10

Table 5b. Comparison of PR and CG algorithms and various EBE approximate factorizations. In each case $\tilde{\underset{\sim}{A}} = \underset{\sim}{A}$.

Algorithm: PR + LS (no BFGS)

EBE approx. fact.	$\log_{10}(-P_f)^{(\dagger)}$	ave. it's. per step	
		steps 1-20	steps 21-50(‡)
symm. Gauss-Seidel	- 13.4	5.41	10
Marchuk	- 15.4	5.03	10
Cholesky	- 15.8	3.95	10
Crout	- 15.1	3.95	10

Algorithm: CG

EBE approx. fact.	$\log_{10}(-P_f)^{(\dagger)}$	ave. it's. per step	
		steps 1-20	steps 21-50(§)
symm. Gauss-Seidel	- 25.3	3.95	9.0
Marchuk	- 25.3	3.50	8.3
Cholesky	- 25.3	3.45	7.9
Crout	- 25.3	2.95	8.0

Notes:

(†) P_f, the final value of potential energy, is minimized by the exact steady-state solution. Consequently, the more negative $\log_{10}(-P_f)$, the better the approximation of the steady-state.

(‡) The maximum number of iterations was 10. The PR algorithm failed to converge for steps 21-50.

(§) The CG algorithm converged in less than or equal to 10 iterations in all cases.

5c. Fluid Mechanics

Two of us (T.J.R.H. and T.E.T.) have recently been engaged in research on the calculation of compressible inviscid flows using the Euler equations. For background on the finite element procedures employed, consult [H18, T1]. A pilot study was performed to assess the feasibility of the EBE approximate factorization procedure in this context. The PR algorithm was employed with line searches based upon (2.2), but no BFGS vectors. The basic Marchuk EBE factorization, (3.23), was employed. Note that the $\underline{A}$ matrix is unsymmetric and potentially indefinite in these applications.

We view the following results as encouraging. However, we believe significant improvement can be made by use of a different driver algorithm and Crout EBE factorizations. We hope to pursue this in future work.

The problem considered was the flow around a circular cylinder. The computational domain and boundary conditions are shown in Figure 12. The free stream parameters are:

$$\rho_\infty = 1$$

$$u_\infty = M_\infty$$

$$v_\infty = 0$$

$$c_\infty = 1$$

The notation is as follows: ρ is the density; u and v are the horizontal and vertical velocity components; M is the Mach number; c is the acoustic speed; and the subscript ∞ indicates a free-stream value. The above data enable determination of the total specific energy e_∞. The free stream values of ρ, u, v and e are employed as initial conditions. The free stream Mach number determines the character of the flow. Two cases were considered:

$$M_\infty = \begin{cases} 0.3 & , \text{ subsonic} \\ 0.5 & , \text{ transonic} \end{cases}$$

The finite element mesh is shown in Figure 13. There are 336 bilinear elements. About the cylinder, 9 elements are in the radial direction and 32 are in the circumferential direction. The time steps employed were

$$\Delta t = \begin{cases} .30 & , \text{ subsonic} \\ .26 & , \text{ transonic} \end{cases}$$

Estimated critical time steps were

$$\Delta t = \begin{cases} .130 & , \text{ subsonic} \\ .075 & , \text{ transonic} \end{cases}$$

resulting in

$$\Delta t/\Delta t_{crit} = \begin{cases} 2.31 & \text{, subsonic} \\ 3.47 & \text{, transonic} \end{cases}$$

The calculations were allowed to run for 20 iterations per time step without a termination criterion in order to get a complete picture of the convergence behavior.

Subsonic case

In this case the solution is smooth and symmetric about the cylinder. Figures 14 and 15 show the Mach number and pressure coefficient about the cylinder at step 20. Convergence information is presented in Figure 16. These figures compare the selections of $\underset{\sim}{W}$. When $\underset{\sim}{W} = \underset{\sim}{D}_S(\underset{\sim}{A})$, the convergence is typically rapid for the first few iterations, then somewhat slow. On the other hand, when $\underset{\sim}{W}$ = the diagonal of the mass matrix, the initial convergence rate is somewhat slower than for the previous case, but subsequently it is much more rapid. A value of $\varepsilon = 1$ was used in these calculations. It is conjectured that a larger value might have further improved convergence. The superiority of the selection of $\underset{\sim}{W}$ as the diagonal of the mass over $\underset{\sim}{W} = \underset{\sim}{D}_S(\underset{\sim}{A})$ is shown even more clearly in the transonic case.

Transonic case

For this problem the choice $\underset{\sim}{W} = \underset{\sim}{D}_S(\underset{\sim}{A})$ fails to converge (see Figure 17). The diagonal mass has some difficulty converging in the first time step, however, the convergence is fairly rapid thereafter. The Mach number and velocity vectors are presented at step 10 in Figures 18 and 19, respectively. The Mach number profile behind the shock has a slight oscillation due to the coarseness of the mesh and particular transient algorithm employed. Superior results may be obtained for this problem and these will be reported upon in future work.

6. Conclusions

In this paper a variety of EBE approximate factorization techniques have been proposed and compared on test problems. In the context of symmetric, positive definite heat conduction operators the CG algorithm performed better than the PR algorithm.

The PR algorithm with BFGS updates performed well in nonlinear structural problems. However, the need to store BFGS vectors is considered a serious disadvantage in the present context, and thus the fixed storage requirements of the CG algorithm renders it preferable.

Among the EBE factorizations compared, the Crout variant seemed best. However, the Cholesky, Marchuk, and symmetrized Gauss-Seidel versions also were effective and thus a preference for one over another may need to be based on other computational considerations.

Results for unsymmetrical problems are vary preliminary. We assume a better driver algorithm than PR can be found for the unsymmetric case. Furthermore, the various EBE factorizations still need to be compared in this context.

The heat conduction calculations comparing parallel and sequential orderings are very exciting. The preliminary indications are that parallel processing with EBE factorizations may be a very efficient computational strategy.

It should be kept in mind that the EBE concept has been explored herein as a finite element, linear equation solving procedure. Although initial attempts at directly using EBE ideas to develop time stepping algorithms had some deficiencies it may still ultimately prove profitable to couple EBE concepts with the time-stepping loop and even the nonlinear iterative loop. It is interesting to note that the multigrid method found its initial success as a linear equation solver, but in the most recent and successful variants the multigrid philosophy permeates all aspects of the methodology (see Brandt [B5]).

In problems in which analytically-derived tangent arrays are difficult or impossible to obtain, such as in oil-reservoir simulation, the quasi-Newton method may be employed on the element level [G4] to derive approximate tangents. This process has been used with the basic CG algorithm, but improved results could probably be obtained by employing EBE approximate factorizations as a preconditioner.

It would also appear that the EBE concept could be fruitfully exploited in eigenvalue calculations.

A step has been taken in the development of EBE solution of finite element equation systems. A considerable potential exists for the technique, but much research still remains to be done to bring the methods to fruition.

Acknowledgements

We would like to thank the following organizations for providing resources and support for our research: Civil Engineering Laboratory, Port Hueneme, California; Lockheed Palo Alto Research Laboratory, Palo Alto, California; NASA Ames Research Center, Moffet Field, California; NASA Langley Research Center, Langley, Virginia; and the NASA Lewis Research Center; Cleveland, Ohio.

We would also like to thank the following individuals: H. Adelman, C. Chamis, J. Crawford, H. Lomax, R. Murtha, and G. Olsen.

Some of the results reported upon in Section 5 were obtained in collaboration with K. C. Park.

References

B1. A. J. Baker, "Research on a Finite Element Algorithm for the Three-dimensional Navier-Stokes Equations," AFWAL-TR-82-3012, Wright-Patterson Air Force Base, Ohio, 1982.

B2. T. Belytschko and W. K. Liu, "On Reduced Matrix Inversion for Operator Splitting Methods," preprint.

B3. T. Belytschko and R. Mullen, "Mesh Partitions of Explicit-Implicit Time Integration," Formulations and Computational Algorithms in Finite Element Analysis, Eds. K. J. Bathe et al., M.I.T. Press, Cambridge, Massachusetts, 1977.

B4. T. Belytschko and R. Mullen, "Stability of Explicit-Implicit Mesh Partitions in Time Integration," International Journal for Numerical Methods in Engineering, Vol. 12, No. 10, 1575-1586 (1979).

B5. A. Brandt, "Guide to Multigrid Development," in Multigrid Methods (W. Hackbusch and U. Trottenberg, eds.) Springer-Verlag, Berlin-Heidelberg-New York, 1982.

C1. D. R. Chapman, "Computational Aerodynamics Development and Outlook," AIAA J., Vol. 17, No. 12, 1293-1313, December 1979.

D1. J. E. Dendy and G. Fairweather, "Alternating-direction Galerkin Methods for Parabolic and Hyperbolic Problems on Rectangular Polygons," SIAM J. Numer. Anal., Vol. 2, 144-163 (1975).

D2. J. E. Dennis and J. J. More, "Quasi-Newton Methods: Motivation and Theory," SIAM Review, Vol. 19, No. 1, 46-89, January 1977.

D3. J. Douglas, "On the Numerical Integration of $u_{xx} + u_{yy} = u_t$ by Implicit Methods," J. Soc. Indust. Appl. Math., Vol. 3, 42-65 (1965).

D4. J. Douglas, "Alternating Direction Methods for Three Space Variables," Numer. Math., 4, 41-63 (1962).

D5. J. Douglas and T. Dupont," Galerkin Methods for Parabolic Equations," SIAM J. Numer. Anal., Vol. 7, 575-626 (1970).

D6. J. Douglas and T. Dupont, "Alternating-direction Galerkin Methods on Rectangles," pp. 133-214 in Proceedings Symposium on Numerical Solution of Partial Differential Equations, II, B. Hubbard (ed.), Academic Press, New York, 1971.

D7. J. Douglas and H. H. Rachford, "On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables," Trans. Amer. Math. Soc., Vol. 82, 421-439 (1956).

G1. R. Glowinski, J. Periaux and Q. V. Dinh, "Domain Decomposition Methods for Nonlinear Problems in Fluid Dynamics," INRIA Report No. 147, July 1982.

G2. R. Glowinski, Numerical Methods for Nonlinear Variational Problems, second edition, Springer Series in Computational Physics, Springer-Verlag, New York-Heidelberg-Berlin, to appear.

G3. G. L. Goudreau and R. L. Taylor, "Evaluation of Numerical Integration Methods in Elastodynamics," Computer Methods in Applied Mechanics and Engineering, Vol. 2, 69-97 (1972).

G4. A. Griewank and Ph. L. Toint, "Partitioned Variable Metric Updates for Large Structured Optimization Problems," Numerische Mathematik, Vol. 39, 119-137 (1982).

H1. L. J. Hayes, "A Comparison of Alternating-direction Collocation Methods for the Transport Equation," pp. 169-177 in New Concepts in Finite Element Analysis, AMD Vol. 44, (T.J.R. Hughes et al. eds.), ASME, New York, 1981.

H2. L. J. Hayes, "Finite Element Patch Approximations and Alternating-Direction Methods," Mathematics and Computers in Simulation, Vol. XXII, 25-29 (1980).

H3. L. J. Hayes, "Implementation of Finite Element Alternating-direction Methods on Nonrectangular Regions," International Journal for Numerical Methods in Engineering, Vol. 16, 35-49 (1980).

H4. L. J. Hayes, "Galerkin Alternating-Direction Methods for Nonrectangular Regions Using Patch Approximations," SIAM J. Numer. Anal., Vol. 18, No. 4, 627-643 (1981).

H5. M. R. Hestenes and E. Steifel, "Methods of Conjugate Gradients for Solving Linear Systems," Journal of Research of National Bureau of Standards, Vol. 49, No. 6, 409-436, December 1952.

H6. H. D. Hibbitt and B. I. Karlsson, "Analysis of Pipe Whip," ASME Paper No. 79-PVP-122, presented at the Pressure Vessels and Piping Conference, San Francisco, California, June 25-29, 1979.

H7. H. M. Hilber, Analysis and Design of Numerical Integration Methods in Structural Dynamics, Report No. EERC76-29, Earthquake Engineering Research Center, University of California, Berkeley, California, November 1976.

H8. H. M. Hilber and T.J.R. Hughes, "Collocation, Dissipation, and 'Overshoot' for Time Integration Schemes in Structural Dynamics," Earthquake Engineering and Structural Dynamic, Vol. 6, 99-118 (1978).

H9. H. M. Hilber, T.J.R. Hughes, and R. L. Taylor, "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics: Earthquake Engineering and Structural Dynamics, Vol. 5, 283-292 (1977).

H10. T.J.R. Hughes, "Implicit-explicit Finite Element Techniques for Symmetric and Non-symmetric Systems," pp. 255-267 in Recent Advances in Non-linear Computational Mechanics, (eds. E. Hinton, D.R.J. Owen and C. Taylor) Pineridge Press, Swansea, U.K., 1982.

H11. T.J.R. Hughes, "Analysis of Transient Algorithms with Particular Reference to Stability Behavior," Computational Methods in Transient Analysis, Eds. T. Belytschko and T.J.R. Hughes, North-Holland Publishing Co., Amsterdam (to appear).

H12. T.J.R. Hughes, I. Levit and J. Winget, "Implicit, Unconditionally Stable, Element-by-element Algorithms for Heat Conduction Analysis," Journal of the Engineering Mechanics Division, ASCE, to appear.

H13. T.J.R. Hughes, I. Levit and J. Winget, "An Element-by-element Solution Algorithm for Problems of Structural and Solid Mechanics," Computer Methods in Applied Mechanics and Engineering, to appear.

H14. T.J.R. Hughes and W. K. Liu, "Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory," Journal of Applied Mechanics, Vol. 45, 371-374 (1978).

H15. T.J.R. Hughes and W. K. Liu, "Implicit-Explicit Finite Elements in Transient Analysis: Implementation and Numerical Examples," Journal of Applied Mechanics, Vol. 45, 375-378 (1978).

H16. T.J.R. Hughes, K. S. Pister, and R. L. Taylor, "Implicit-Explicit Finite Elements in Nonlinear Transient Analysis," Computer Methods in Applied Mechanics and Engineering, Vols. 17/18, 159-182 (1979).

H17. T.J.R. Hughes and R. A. Stephenson, "Convergence of Implicit-Explicit Algorithms in Nonlinear Transient Analysis," International Journal of Engineering Science, Vol. 19, No. 2, 295-302 (1981).

H18. T.J.R. Hughes, T. E. Tezduyar and A. N. Brooks, "A Petrov-Galerkin Finite Element Formulation for Systems of Conservation Laws with Special Reference to the Compressible Euler Equations," Proceedings of the IMA Conference on Numerical Methods in Fluid Dynamics, University of Reading, England, March 1982. To be published by Academic Press. ← *Numerical Methods for Fluid Dynamics (eds. K.W. Morton and M.J. Baines), Academic Press, London (1982) 97-125.*

K1. R. D. Krieg and S. W. Key, "Transient Shell Response by Numerical Time Integration," International Journal for Numerical Methods in Engineering, Vol. 7, 273-286 (1973).

K2. R. D. Krieg and D. B. Krieg, "Accuracies of Numerical Solution Methods for the Elastic-Perfectly Plastic Model," Journal of Pressure Vessel Technology, 510-515, November 1977.

M1. G. I. Marchuk, Methods of Numerical Mathematics, Springer-Verlag, New York-Heidelberg-Berlin, 1975.

N1. N. M. Newmark, "A Method of Computation for Structural Dynamics," Journal of the Engineering Mechanics Division, ASCE, 67-94 (1959).

N2. B. Nour-Omid and B. N. Parlett, "Element Preconditioning," Report PAM-103, Center for Pure and Applied Mathematics, University of California, Berkeley, October 1982.

O1. M. Ortiz, P. M. Pinsky and R. L. Taylor, "Unconditionally Stable Element-by-element Algorithms for Dynamic Problems", Computer Methods in Applied Mechanics and Engineering, in press.

P1. C. C. Paige and M. A. Saunders, "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares," ACM Transactions on Mathematical Software, Vol. 8, No.1, 43-71, March 1982.

P2. K. C. Park, "Semi-Implicit Transient Analysis Procedure for Structural Dynamics Analysis," Int. J. for Num. Meth. in Engng., 18, 604-622 (1982).

P3. D. W. Peaceman and H. H. Rachford, "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. Soc. Indust. Appl. Math., Vol. 3, 28-41 (1955).

R1. R. S. Rogallo, "Numerical Experiments in Homogeneous Turbulence," NASA TN 81315, September 1981.

T1. T.E. Tezduyar and T.J.R. Hughes, "Development of Time-accurate Finite Element Techniques for First-order Hyperbolic Systems with Particular Emphasis on the Compressible Euler Equations" Final Report, NASA-Ames University Consortium Interchange No. NCA2-OR745-104, April 1982.

T2. F. Thomasset, Implementation of Finite Element Methods for Navier-Stokes Equations, Springer-Verlag, New York-Heidelberg-Berlin, 1981.

T3. S. Timoshenko and J. N. Goodier, Theory of Elasticity, Second Edition, McGraw-Hill, New York, 1951.

T4. D. M. Trujillo, "An Unconditionally Stable, Explicit Algorithm for Finite-Element Heat Conduction Analysis", Journal of Nuclear Engineering and Design, 41, 175-180, 1977.

T5. D. M. Trujillo, "An Unconditionally Stable, Explicit Algorithm for Structural Dynamics," International Journal for Numerical Methods in Engineering, Vol. 11, 1579-1592 (1977).

W1. R. F. Warming and R. M. Beam, "On the Construction and Application of Implicit Factored Schemes for Conservation Laws." SIAM-AMS PROCEEDINGS, Vol. 11, 85-129 (1978).

W2. J. Winget, Ph.D. Thesis, California Institute of Technology, Pasadena, forthcoming.

Y1. N. N. Yanenko, The Method of Fractional Steps, Springer-Verlag, New York-Heidelberg-Berlin, 1971.

Appendix I - Derivation of Linear Algebraic Systems in the Finite Element Analysis of Nonlinear Mechanics Problems

Semi-discrete Equations of Nonlinear Mechanics

Consider the following semi-discrete system

$$\underset{\sim}{M}\,\underset{\sim}{a} = \underset{\sim}{F} \tag{I.1}$$

where $\underset{\sim}{M}$, $\underset{\sim}{a}$ and $\underset{\sim}{F}$ represent the (generalized) mass matrix, acceleration vector and force vector, respectively. Equation (I.1) may be thought of as arising from a finite element discretization of a solid, fluid, structure or combined system. In general, $\underset{\sim}{M}$, $\underset{\sim}{a}$ and $\underset{\sim}{F}$ each depend on time (t). Explicit characterization of $\underset{\sim}{M}$, $\underset{\sim}{a}$ and $\underset{\sim}{F}$ may be given for particular systems under consideration.

Nonlinear Structural and Solid Mechanics

In nonlinear structural and solid mechanics the Lagrangian kinematical description is frequently adopted. In this case the important kinematical quantities are $\underset{\sim}{d}$, the material-particle displacement from a reference configuration; $\underset{\sim}{v} = \dot{\underset{\sim}{d}}$, the particle velocity; and $\underset{\sim}{a} = \dot{\underset{\sim}{v}} = \ddot{\underset{\sim}{d}}$, the particle acceleration. Dots indicate the Lagrangian time-derivative in which the material particle is held fixed. The forces are assumed to take the form

$$\underset{\sim}{F} = \underset{\sim}{F}^{ext} - \underset{\sim}{N} \tag{I.2}$$

where $\underset{\sim}{F}^{ext}$ is the vector of given external forces and $\underset{\sim}{N}$ denotes the vector of internal forces, which may depend upon $\underset{\sim}{d}$, $\dot{\underset{\sim}{d}}$ and their histories. To make the dependence precise, one need introduce equations which define the constitutive (i.e. stress-deformation) behavior of the materials in question. These equations vary widely in type and complexity. For example, they may be algebraic equations, differential equations or integro-differential equations. In addition, inequality constraints may be present, such as in plasticity theory.

Time Discretization

To solve the semi-discrete problem, a time-discretization algorithm needs to be introduced. For purpose of illustration we shall employ the Newmark family of methods [N1]. Generalization to other time integrators, such as the Hilber-Hughes-Taylor algorithm [H6, H8, H9, H11] which possesses improved properties, may be easily facilitated without essential alteration to the following formulation.

The Newmark "predictors" are given

$$\tilde{\underset{\sim}{d}}_{n+1} = \underset{\sim}{d}_n + \Delta t\, \underset{\sim}{v}_n + \frac{\Delta t^2}{2}(1 - 2\beta)\underset{\sim}{a}_n \tag{I.3}$$

$$\tilde{\underset{\sim}{v}}_{n+1} = \underset{\sim}{v}_n + \Delta t\,(1 - \gamma)\underset{\sim}{a}_n \tag{I.4}$$

where subscripts refer to the step number; Δt is the time step; $\underset{\sim}{d}_n$, $\underset{\sim}{v}_n$ and $\underset{\sim}{a}_n$ are the approximations to $\underset{\sim}{d}(t_n)$, $\dot{\underset{\sim}{d}}(t_n)$ and $\ddot{\underset{\sim}{d}}(t_n)$, respectively; and β and γ are parameters which govern the accuracy and stability of the method [G3, H7, K1].

Calculations commence with the given initial data (i.e., $\underset{\sim}{d}_0$ and $\underset{\sim}{v}_0$) and $\underset{\sim}{a}_0$ which may be calculated from

$$\underset{\sim}{M}\,\underset{\sim}{a}_0 = \underset{\sim}{F}^{ext}_0 - \underset{\sim}{N}_0 \tag{I.5}$$

If $\underset{\sim}{M}$ is diagonal, as is common in structural dynamics, the solution of (I.5) is rendered trivial. Otherwise, a factorization, forward reduction and back substitution are necessary to obtain $\underset{\sim}{a}_0$.

In the sequel we are only interested in members of the Newmark family for which $\beta > 0$.

In each time step a nonlinear algebraic problem arises which may be solved by Newton-Raphson and quasi-Newton-type iterative procedures. There are several ways of going about this. In the following implementation an algebraic problem is formulated in which acceleration increments are the unknowns. This form of the implementation is useful in that disparate field theories, such as fluid mechanics and heat conduction, may be formally considered as special cases. We shall return to this point later on.

acceleration formulation

$$i = 0 \qquad \text{(i is the iteration counter)} \tag{I.6}$$

$$\left.\begin{aligned} \underset{\sim}{d}^{(i)}_{n+1} &= \tilde{\underset{\sim}{d}}_{n+1} && (I.7)\\ \underset{\sim}{v}^{(i)}_{n+1} &= \tilde{\underset{\sim}{v}}_{n+1} && (I.8)\\ \underset{\sim}{a}^{(i)}_{n+1} &= \underset{\sim}{0} && (I.9) \end{aligned}\right\} \text{(predictor phase)}$$

$$\underset{\sim}{R} = \underset{\sim}{F}^{ext}_{n+1} - \underset{\sim}{N}^{(i)}_{n+1} - \underset{\sim}{M}^{(i)}_{n+1}\,\underset{\sim}{a}^{(i)}_{n+1} \quad \text{(residual, or out-of-balance, force)} \tag{I.10}$$

$$\underset{\sim}{M}^{*} = \underset{\sim}{M}^{(i)}_{n+1} + \gamma\Delta t\,\underset{\sim}{C}^{(i)}_{n+1} + \beta\Delta t^2\,\underset{\sim}{K}^{(i)}_{n+1} \quad \text{(effective mass)} \tag{I.11}$$

$$\boxed{\underset{\sim}{M}^{*}\,\Delta\underset{\sim}{a} = \underset{\sim}{R}} \tag{I.12}$$

$$\left.\begin{aligned} \underset{\sim}{a}^{(i+1)}_{n+1} &= \underset{\sim}{a}^{(i)}_{n+1} + \Delta\underset{\sim}{a} && (I.13)\\ \underset{\sim}{v}^{(i+1)}_{n+1} &= \underset{\sim}{v}^{(i)}_{n+1} + \gamma\Delta t\,\Delta\underset{\sim}{a} && (I.14)\\ \underset{\sim}{d}^{(i+1)}_{n+1} &= \underset{\sim}{d}^{(i)}_{n+1} + \beta\Delta t^2\,\Delta\underset{\sim}{a} && (I.15) \end{aligned}\right\} \text{(corrector phase)}$$

If additional iterations are to be performed, i is replaced by i+1 , and calculations resume with (I.10). Either a fixed number of iterations may be performed, or iterating may be terminated when $\Delta\underset{\sim}{a}$ and/or $\underset{\sim}{R}$ satisfy preassigned convergence conditions. When the iterative phase is completed, the solution at step n+1 is defined by the last iterates (viz. $\underset{\sim}{d}_{n+1} = \underset{\sim}{d}^{(i+1)}_{n+1}$; $\underset{\sim}{v}_{n+1} = \underset{\sim}{v}^{(i+1)}_{n+1}$; and $\underset{\sim}{a}_{n+1} = \underset{\sim}{a}^{(i+1)}_{n+1}$). At this point, n is replaced by n+1 , and

calculations for the next time step may begin.

In practice, $\underset{\sim}{d}_n$, $\underset{\sim}{v}_n$ and $\underset{\sim}{a}_n$ are generally saved during the iterative phase along with $\underset{\sim}{a}_{n+1}^{(i+1)}$; but $\underset{\sim}{v}_{n+1}^{(i+1)}$ and $\underset{\sim}{d}_{n+1}^{(i+1)}$ may be computed as needed, on the element level.

The matrices $\underset{\sim}{C}$ and $\underset{\sim}{K}$ are the tangent damping and tangent stiffness matrices, respectively. These are linearized operators associated with $\underset{\sim}{N}$. For example, if $\underset{\sim}{N}$ is an algebraic function of $\underset{\sim}{d}$ and $\dot{\underset{\sim}{d}}$, then

$$\underset{\sim}{K} = \partial \underset{\sim}{N} / \partial \underset{\sim}{d} \tag{I.16}$$

and

$$\underset{\sim}{C} = \partial \underset{\sim}{N} / \partial \dot{\underset{\sim}{d}} \tag{I.17}$$

Generally in structural and solid mechanics, $\underset{\sim}{M}$, $\underset{\sim}{K}$ and $\underset{\sim}{C}$ are symmetric, $\underset{\sim}{M}$ and $\underset{\sim}{K}$ are positive-definite, and $\underset{\sim}{C}$ is positive semi-definite.

So-called implicit-explicit mesh partitions [B3, B4, H10, H11, H14-H17] may be encompassed by the above formulation simply by excluding explicit element/node contributions from the definitions of $\underset{\sim}{C}$ and $\underset{\sim}{K}$. A totally explicit formulation is attained by ignoring $\underset{\sim}{C}$ and $\underset{\sim}{K}$. In these cases it is necessary to employ a diagonal mass matrix in explicit regions to attain full computational efficiency.

It may be observed that the preceding algorithm may be specialized to nonlinear statics and linear dynamics and statics:

nonlinear statics

In this case ignore $\underset{\sim}{M}$ and $\underset{\sim}{C}$ and set $\underset{\sim}{v}$ and $\underset{\sim}{a}$ to zero throughout.

linear dynamics

In this case $\underset{\sim}{M}$, $\underset{\sim}{C}$ and $\underset{\sim}{K}$ are constant and

$$\underset{\sim}{N} = \underset{\sim}{C}\,\underset{\sim}{v} + \underset{\sim}{K}\,\underset{\sim}{d} \tag{I.18}$$

linear statics

In this case ignore $\underset{\sim}{M}$ and $\underset{\sim}{C}$, set $\underset{\sim}{v}$ and $\underset{\sim}{a}$ to zero throughout, $\underset{\sim}{K}$ is constant and

$$\underset{\sim}{N} = \underset{\sim}{K}\,\underset{\sim}{d} \tag{I.19}$$

Fluid Mechanics and Heat Conduction

The preceding formalism also subsumes other physical theories such as fluid machanics and heat conduction. As an example, in fluid mechanics we will consider the form the algorithm takes for the compressible Euler equations. (For further details on this topic consult [H18, T1].) Both the compressible Euler equations and heat conduction lead to first-order semi-discrete systems. Thus, in the preceding all terms emanating from the appearance of $\underset{\sim}{d}$ as an argument of $\underset{\sim}{N}$ may be omitted. For the compressible Euler equations $\underset{\sim}{v}$ is viewed as a state vector which contains nodal density, momenta and energy degrees-of-freedom, and $\underset{\sim}{a}$ is viewed as the Eulerian time derivative of $\underset{\sim}{v}$. The coefficient matrix $\underset{\sim}{M}^*$ will generally be unsymmetric. In heat conduction, $\underset{\sim}{v}$ is the vector

of nodal temperatures. $\underset{\sim}{M}^*$ is usually symmetric for this case. The final algorithm specializes as follows:

$$i = 0 \qquad \text{(i is the iteration counter)} \tag{I.20}$$

$$\left.\begin{aligned} \underset{\sim}{v}_{n+1}^{(i+1)} &= \underset{\sim}{\tilde{v}}_{n+1} \qquad (I.21)\\ \underset{\sim}{a}_{n+1}^{(i)} &= \underset{\sim}{0} \qquad (I.22) \end{aligned}\right\} \text{(predictor phase)}$$

$$\underset{\sim}{R} = \underset{\sim}{F}_{n+1}^{ext} - \underset{\sim}{N}_{n+1}^{(i)} - \underset{\sim}{M}_{n+1}^{(i)}\, \underset{\sim}{a}_{n+1}^{(i)} \tag{I.23}$$

$$\underset{\sim}{M}^* = \underset{\sim}{M}_{n+1}^{(i)} + \gamma\Delta t\, \underset{\sim}{C}_{n+1}^{(i)} \tag{I.24}$$

$$\boxed{\underset{\sim}{M}^*\, \Delta\underset{\sim}{a} = \underset{\sim}{R}} \tag{I.25}$$

$$\left.\begin{aligned} \underset{\sim}{a}_{n+1}^{(i+1)} &= \underset{\sim}{a}_{n+1}^{(i)} + \Delta\underset{\sim}{a} \qquad (I.26)\\ \underset{\sim}{v}_{n+1}^{(i+1)} &= \underset{\sim}{v}_{n+1}^{(i)} + \gamma\Delta t\, \Delta\underset{\sim}{a} \qquad (I.27) \end{aligned}\right\} \text{(corrector phase)}$$

To simplify the writing in the body of this paper we adopt the following notations in place of (I.12) and (I.25):

$$\underset{\sim}{A}\, \underset{\sim}{x} = \underset{\sim}{b} \tag{I.28}$$

Thus during each step, at each iteration, we wish to solve (I.28) in which $\underset{\sim}{A}$ is assembled from element arrays, that is

$$\underset{\sim}{A} = \sum_{e=1}^{n_{e\ell}} \underset{\sim}{A}^e \tag{I.29}$$

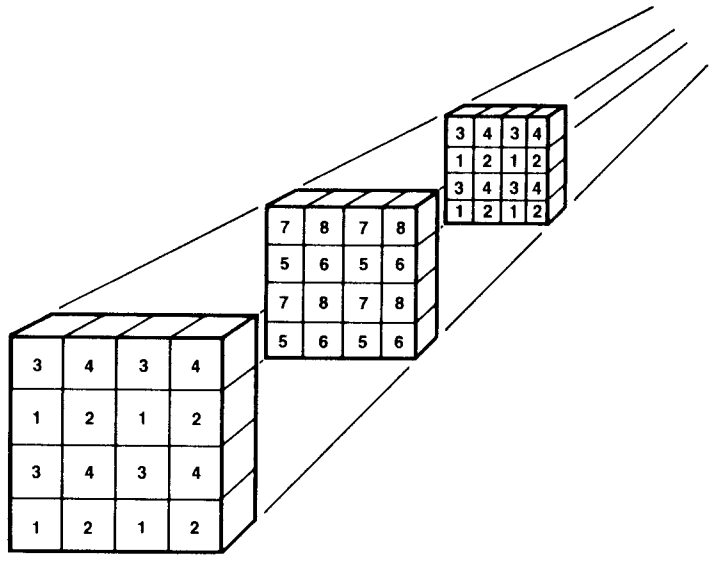


Figure 1. Decomposition of three-dimensional domain into eight groups of brick elements for parallel processing.

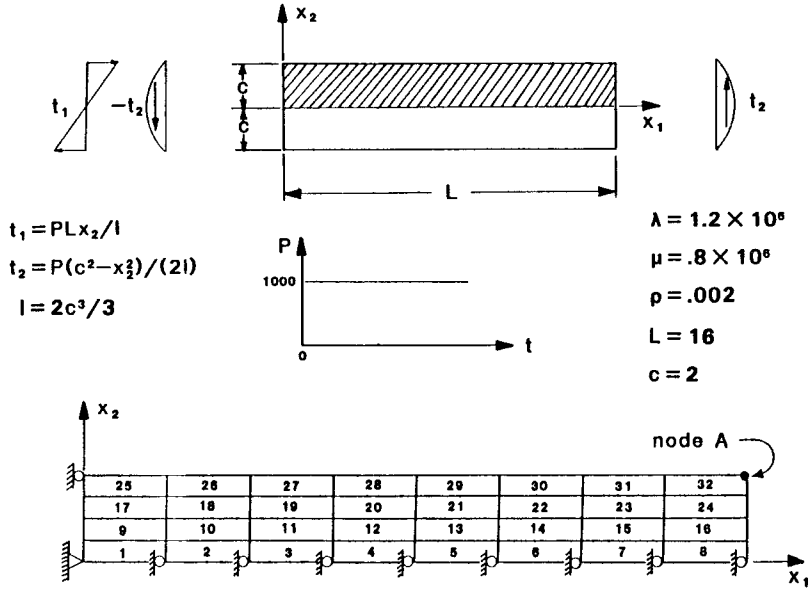


Figure 2. Problem definition and finite element mesh

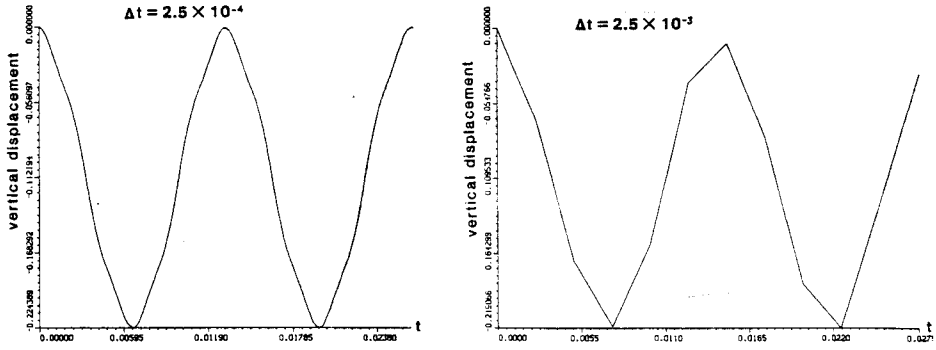


Figure 3. Vertical displacement of node A

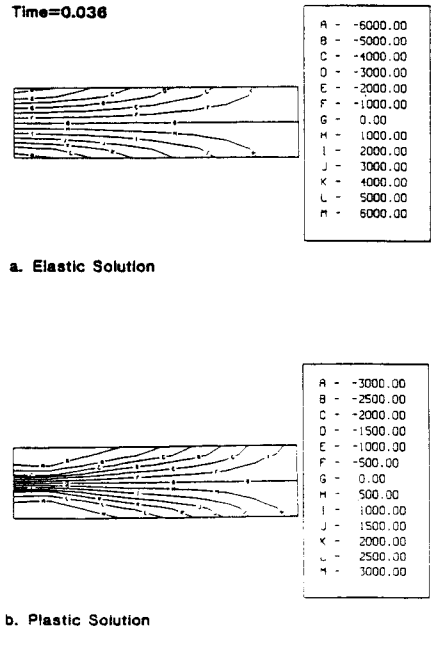


Figure 4. Stress σ_{11}

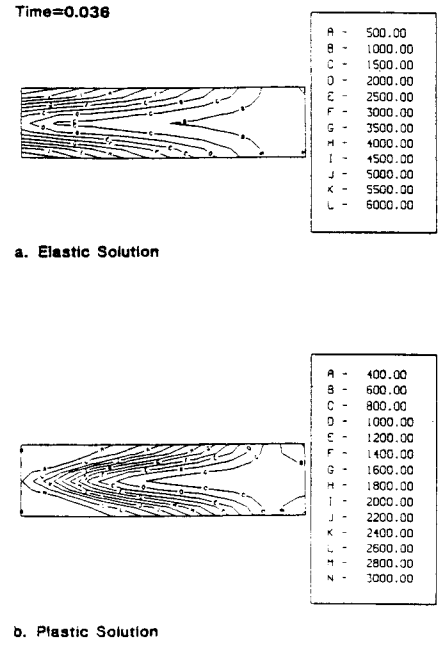
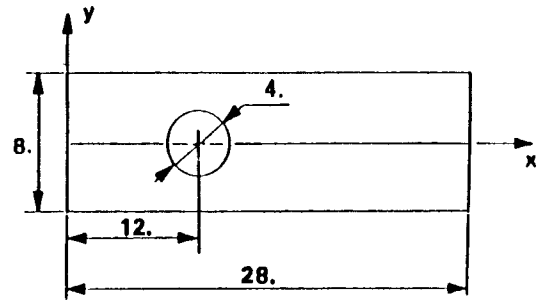
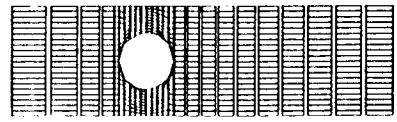


Figure 5. Von Mises Stress



a. Geometry Definition



$\rho = 0.02$

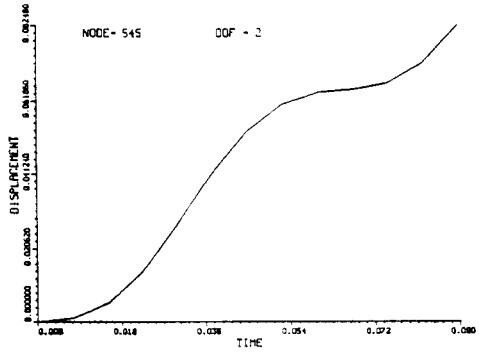
$\lambda = 1200000.$

$\mu = 800000.$

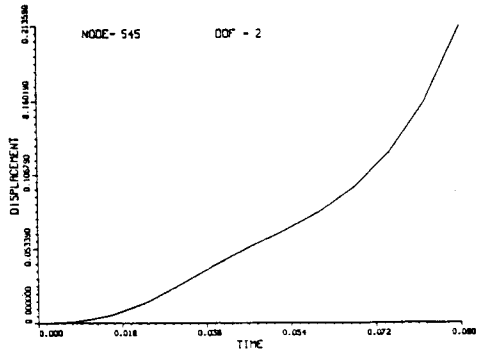
$\sigma y = 1000.$

b. Finite Element Mesh

Figure 6



a. Elastic Solution



b. Plastic Solution

Figure 7. Displacement time history

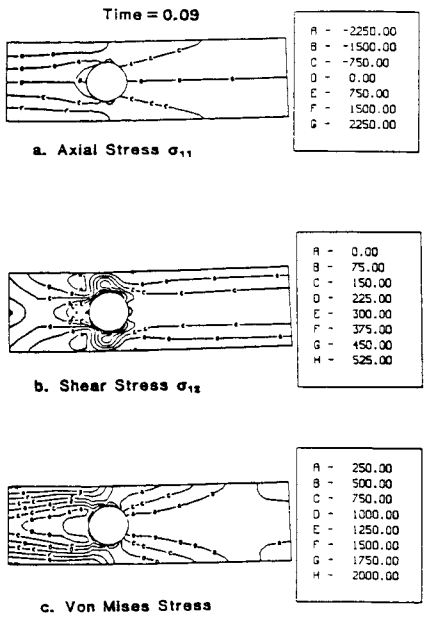


Figure 8. Stress Distribution (elastic solution)

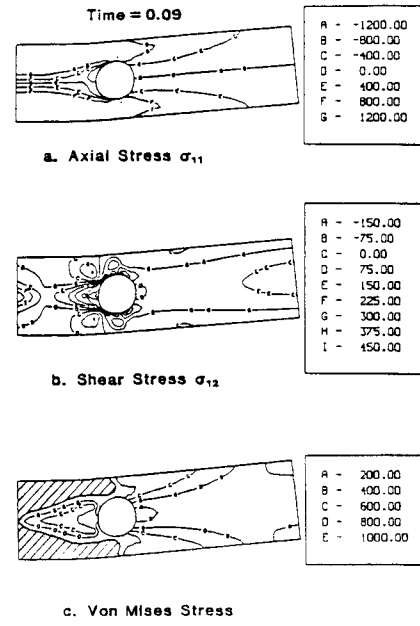
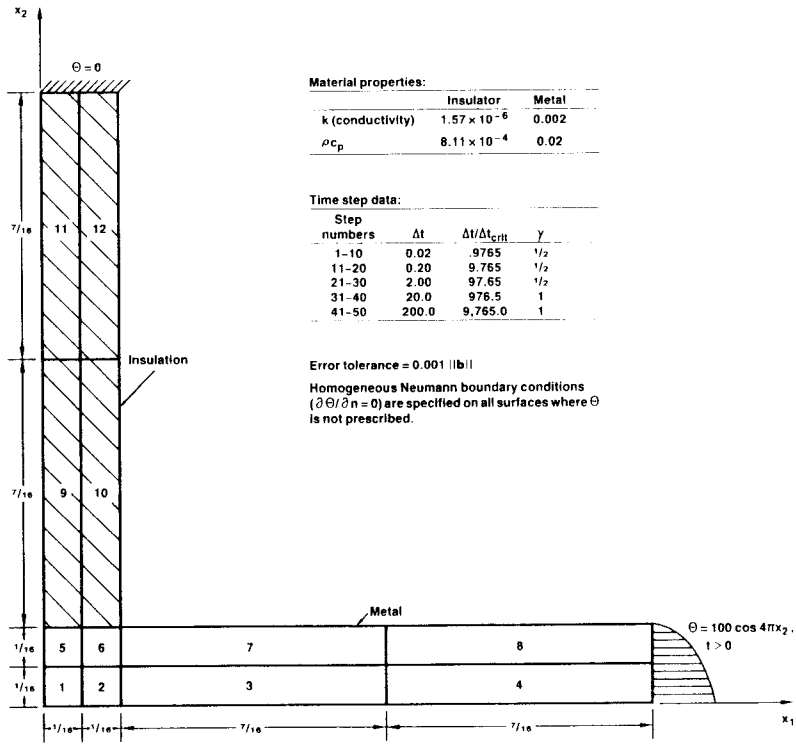


Figure 9. Stress Distribution (plastic solution)



Material properties:

	Insulator	Metal
k (conductivity)	1.57×10^{-6}	0.002
ρc_p	8.11×10^{-4}	0.02

Time step data:

Step numbers	Δt	$\Delta t/\Delta t_{crit}$	γ
1-10	0.02	.9765	1/2
11-20	0.20	9.765	1/2
21-30	2.00	97.65	1/2
31-40	20.0	976.5	1
41-50	200.0	9,765.0	1

Error tolerance = 0.001 ||b||

Homogeneous Neumann boundary conditions ($\partial\Theta/\partial n = 0$) are specified on all surfaces where Θ is not prescribed.

Figure 10. Problem description for NASA insulated structure test problem.

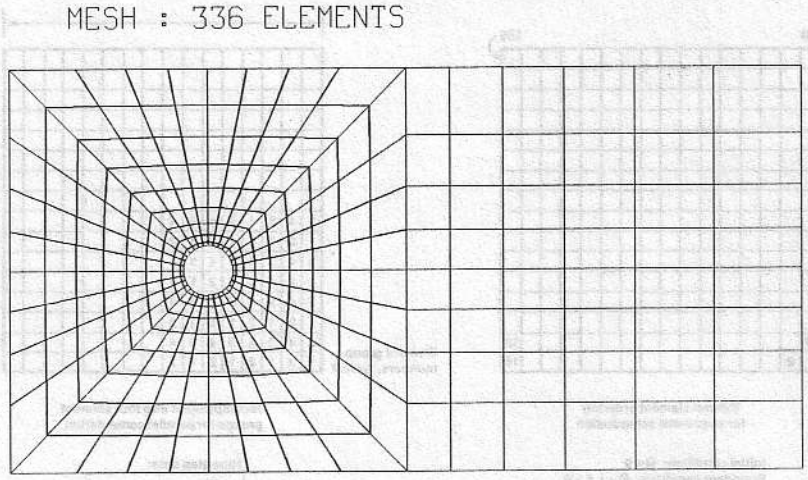


Figure 13

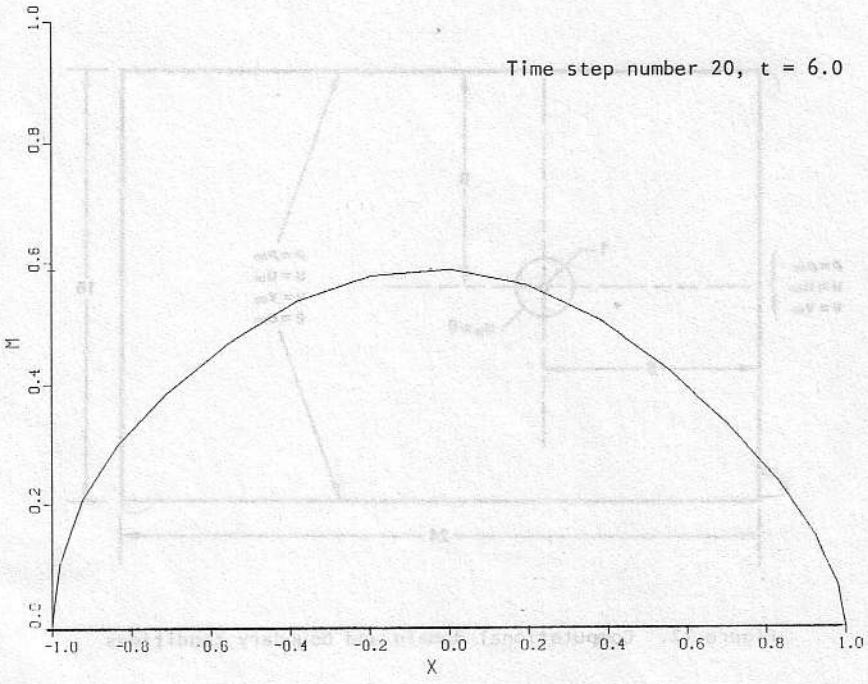


Figure 14. Mach number, subsonic case

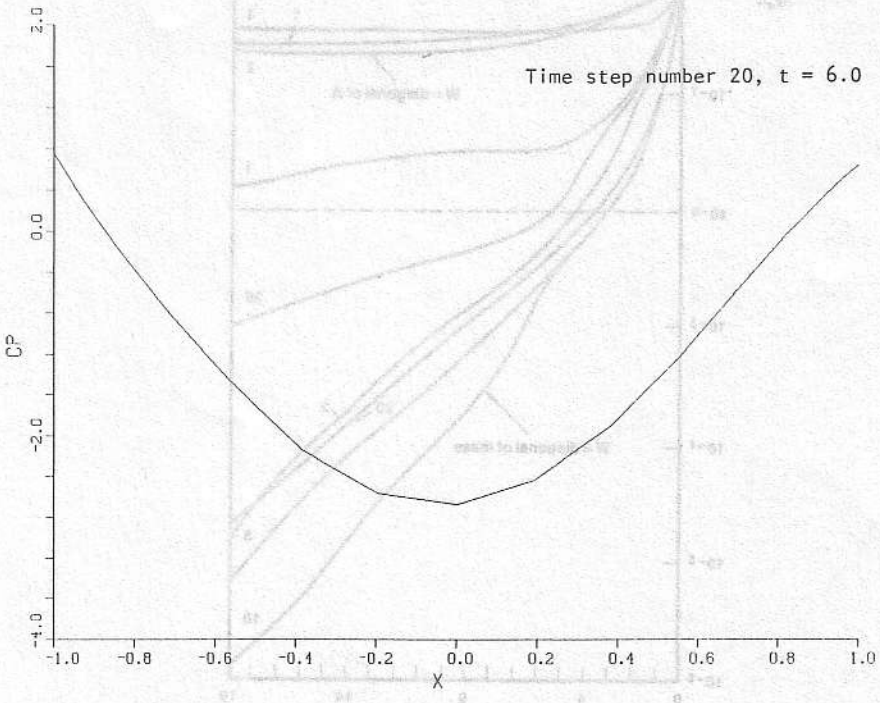


Figure 15. Pressure coefficient, subsonic case

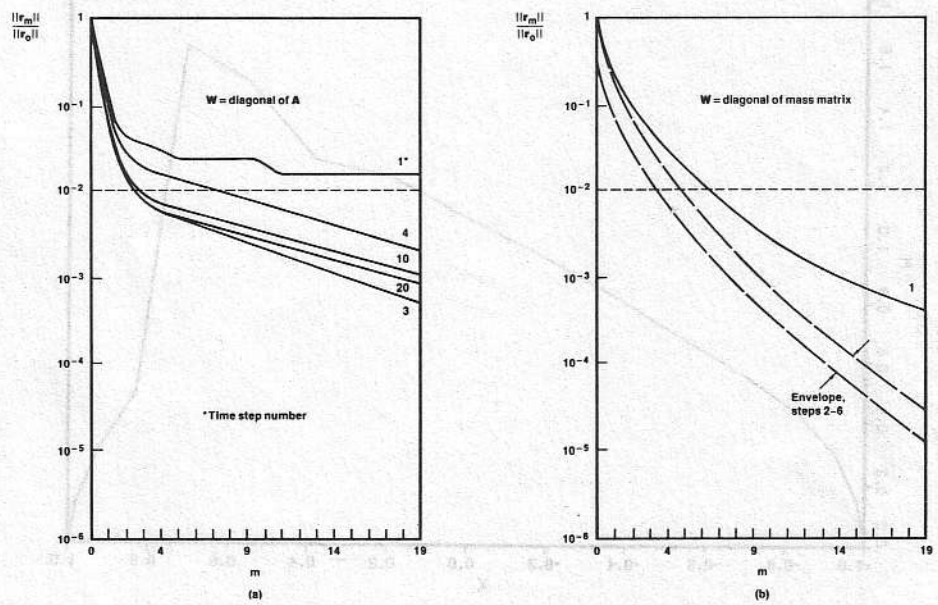


Figure 16. Convergence of residual for subsonic case

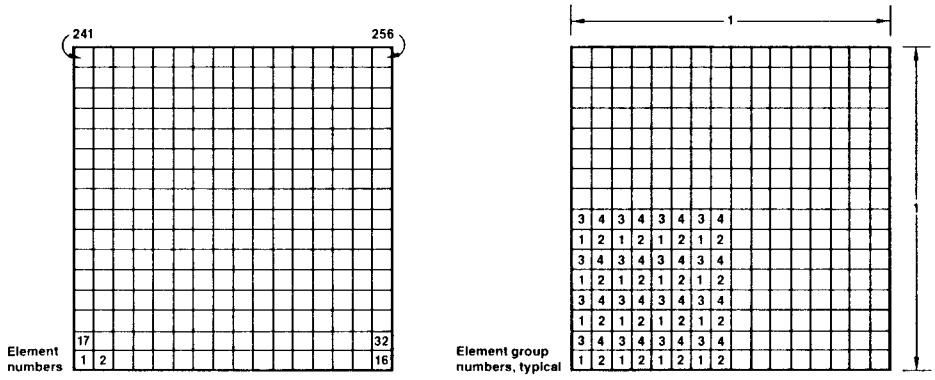


Figure 11. Problem descriptions for parallel/sequential comparison

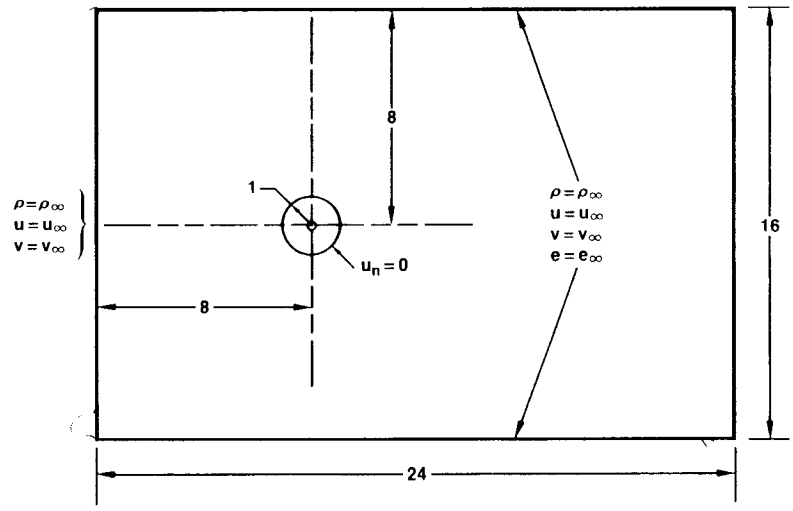


Figure 12. Computational domain and boundary conditions

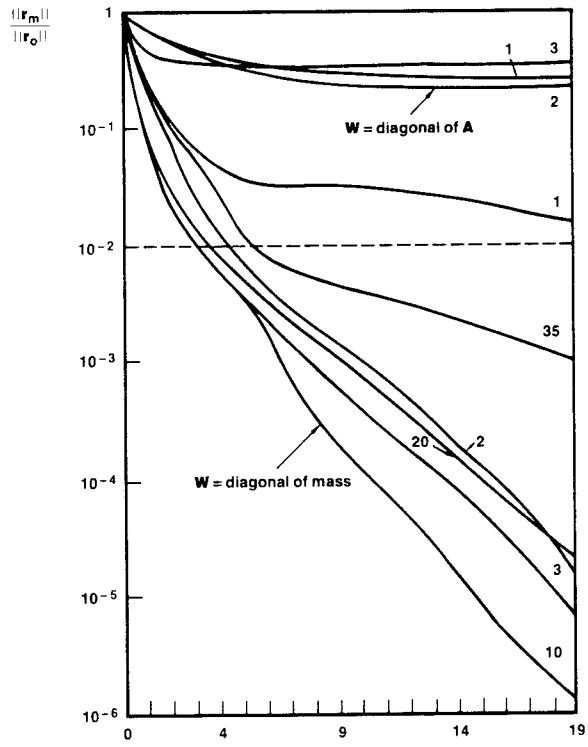


Figure 17. Convergence of residual for transonic case

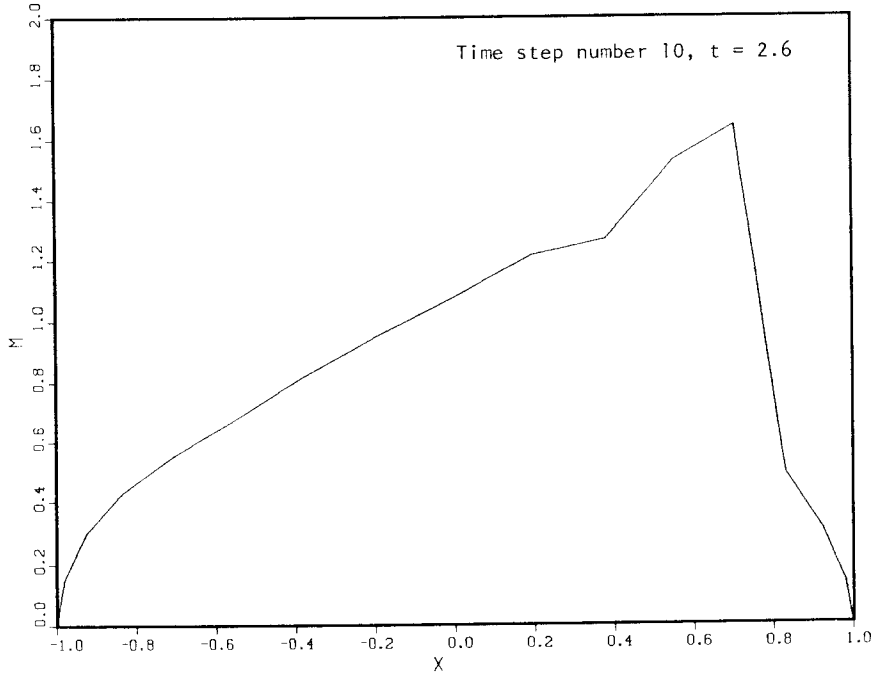


Figure 18. Mach number, transonic case

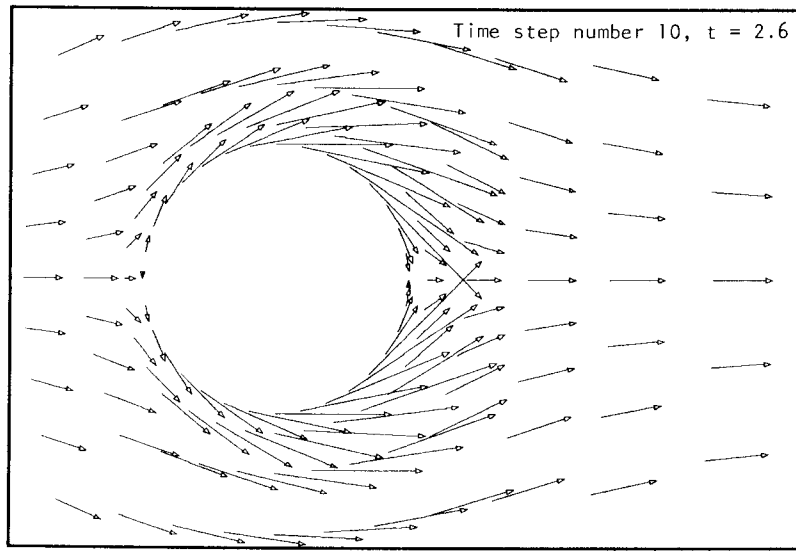


Figure 19. Velocity vectors about cylinder, transonic case