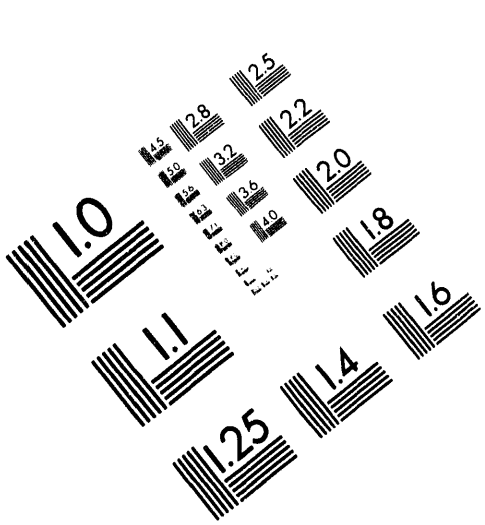
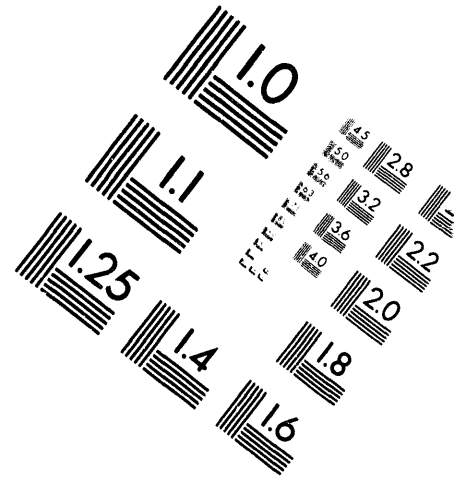
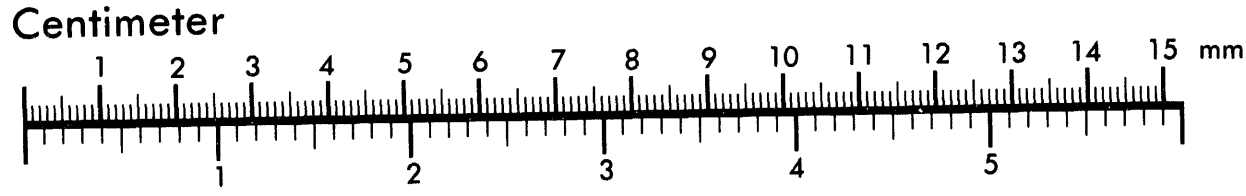


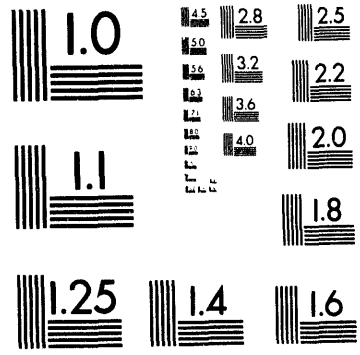
Association for Information and Image Management

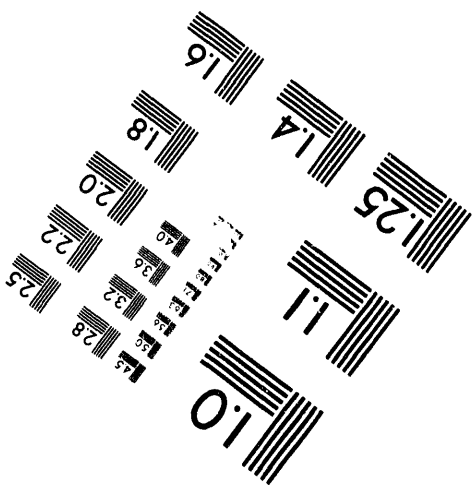
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

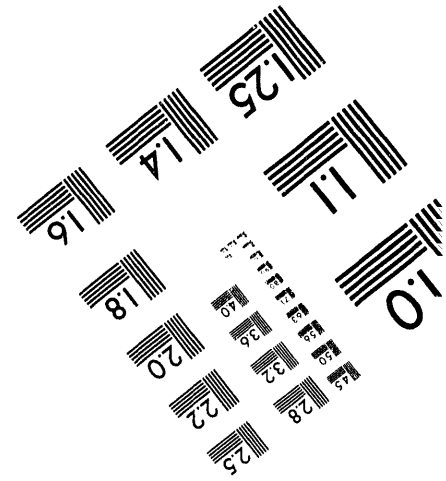
301/587-8202











MANUFACTURED TO AIIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 1

SAIC-94/1098:APPAT-163
May 9, 1994

TRANSPORT AND DYNAMICS IN TOROIDAL FUSION SYSTEMS

REPORT OF SECOND YEAR PROGRESS 1993-1994

Science Applications International Corporation
An Employee-Owned Company

Worked Performed Under
DOE Grant DE-FG03-91ER54124

DALTON D. SCHNACK
PRINCIPAL INVESTIGATOR
Applied Plasma Physics and Technology Division
Science Applications International Corp.
San Diego, CA 92121

10260 Campus Point Drive, San Diego, California 92121 (619) 546-6000
Other SAIC Offices: Albuquerque, Boston, Colorado Springs, Dayton, Huntsville, Las Vegas, Los Angeles, McLean, Oak Ridge, Orlando, Palo Alto, Seattle, and Tucson

TABLE OF CONTENTS

Section		Figure
1.0	INTRODUCTION	1
2.0	COMPUTATIONAL MESHES	4
	2.1 Finite-dimensional Systems and Structured Meshes	4
	2.2 Unstructured Meshes	5
	2.3 Primary and Dual Meshes	7
	2.4 Delaunay Triangles and Voronoi Polygons	8
	2.5 The Barycenter, or Centroid, Dual Mesh	9
	2.6 The Third (Toroidal) Dimension	10
	2.7 Mesh Refinement	11
3.0	APPROXIMATION TO SPATIAL DIFFERENTIAL OPERATORS	12
4.0	THE MHD EQUATIONS: PLACEMENT OF THE VARIABLES ON THE MESH	15
	4.1 Hydrodynamic Variables	16
	4.2 Electromagnetic Variables	17
	4.3 Averages and Interpolation	19
	4.4 Boundary Conditions	21
5.0	TIME INTEGRATION	22
	5.1 Explicit Advance	22
	5.2 Semi-Implicit and Implicit Solutions	24
6.0	APPLICATIONS	27
	6.1 The Hydrodynamic Shock Tube Problem	27
	6.2 The Magnetohydrodynamic Shock Tube Problem	36
	6.3 Toroidal Equilibria	42
	6.4 Nonlinear Evolution of Toroidal Instabilities	46
7.0	SUMMARY AND CONDITIONS	56
REFERENCES		57

TRANSPORT AND DYNAMICS IN TOROIDAL FUSION SYSTEMS

1.0 INTRODUCTION

Over the past two decades, large scale numerical simulation has played an important role in fusion plasma research. Applications of these techniques to fluid plasma models have led to an interpretation of sawtooth[1] and fishbone oscillations[2] in tokamaks, the tokamak major disruption[3], the tilting mode in field-reversed configurations[4], and to a fundamental understanding of the Reversed-field Pinch dynamo[5]. These calculations were generally performed on spatial grids aligned with fixed coordinate directions. As a result, these calculations were often carried out in generic toroidal or cylindrical geometry.

Recently fusion has evolved from a research program to one that is increasingly engineering based, and generic calculations based on simple geometries have become less relevant. Important problems now involve the quantitative prediction of flows and heat transport near limiters and divertors, which often have complex geometry. These phenomena are driven by fluxes of particles and thermal energy that arise from longer spatial scale motions in the plasma core. The core dynamics may in turn be affected by the edge conditions through recycling fluxes. To answer these practical questions, self consistent calculations that include geometric details are required.

Theoretical issues are also affected by the details of the geometry. For example, the poloidal plasma shape can greatly influence the linear stability properties of a fusion plasma, and such calculations are now routinely performed with the actual poloidal plasma geometry accurately represented. This is often accomplished by employing a coordinate system based on the magnetic field lines, which are fixed throughout the calculation. The resulting metric makes the fluid equations quite complicated, but allows the coordinate system to naturally fit the plasma shape.

Coordinate systems based on magnetic fields have several disadvantages in fully nonlinear simulations because of the dynamical nature of the magnetic field. Primary among these is the non uniqueness of the magnetic topology when finite resistivity is included in the model. These coordinate systems also may become singular at magnetic separatrices, which are essential features of modern tokamaks.

It is thus desirable for future simulations to employ a spatial representation that can readily conform to the geometric details of the plasma and its surroundings, and is independent of the magnetic structure. For accuracy, this representation should also be capable of conforming to the dynamical evolution of short spatial scale structures, such as current filaments and edge density gradients, that may

appear spontaneously and require finer spatial resolution than the surrounding environment.

One candidate for a spatial representation with these features is an *unstructured, adaptive mesh.* In such a mesh the mesh points are not constrained to lie along constant coordinate directions. Instead, mesh points are placed on the boundary to conform with the actual geometry of the problem, and distributed in space to maximize the accuracy of the calculation. Thus placed, the points are connected with line elements that form the edges of triangles. These triangles are the Eulerian control volumes that form the basis for the finite representation of the appropriate fluid equations. In the logical data structure that describes the mesh, mesh points (and associated triangles) can easily be added or deleted dynamically based on pre-defined accuracy criteria. The spatial representation can thus adapt to evolving spatial structures without the mesh distortion problems associated with Lagrangian formulations.

Techniques based on unstructured, adaptive meshes have come to maturity in computational fluid dynamics (CFD), where quantitative predictions in real geometry have become essential in the design of aircraft and gas turbine engines[6]. These methods are generally based the solution of a Riemann problem at each triangle interface (edge) to determine the fluxes of energy, mass, and momentum[7]. The simplest extension of the hydrodynamic model that is appropriate for the description of magnetic fusion plasmas is magnetohydrodynamics (MHD).

In this document we describe an extension of these spatial gridding techniques to an MHD model suitable for the description of the dynamics of toroidal fusion devices. Since the dominant MHD modes in these devices have relatively long toroidal wavelength, the toroidal coordinate is approximated with finite Fourier series. The unstructured, triangular mesh is used to describe the details of the poloidal geometry. With some exceptions, the hydrodynamic variables are treated in a manner analogous to that used in CFD. These quantities (mass, energy, and momentum) are volume based densities that satisfy scalar or vector conservation laws. The electromagnetic variables (the magnetic flux density **B** and the electric current density **J**) are area based densities that satisfy pseudo-vector conservation laws, and have no counterpart in fluid dynamics. These variables are also constrained to remain solenoidal. These quantities are represented on the triangular mesh in a new manner that is an extension of that used on rectangular, structured meshes.

In this work we have chosen to solve the primitive (instead of reduced) MHD equations in order to make the resulting codes and techniques more generally applicable to problems beyond the narrow scope of tokamak plasmas. The temporal stiffness problems inherent in this description of tokamak dynamics that motivate the reduced MHD model are addressed here with the semi-implicit method of time integration[8]. Finally, we remark that, while the present work deals strictly with the MHD equations, other volume based fluid descriptions, such as diffusive transport,

could easily be adapted to these techniques and coupled with the description of the electromagnetic field presented here.

This document is organized as follows. In Section 2 we discuss the properties of structured and unstructured meshes, and the data structures useful for describing them. Issues related to the triangulation of an arbitrary set of points in a plane are also discussed. In Section 3 we derive a finite volume approximation to the resistive MHD equations suitable for use on an unstructured, triangular mesh in toroidal geometry. Boundary conditions are discussed here. The specific MHD model, and its implementation on the unstructured mesh, is discussed in Section 4. In Section 5 we discuss methods of time integration, and describe our implementation of semi-implicit and fully implicit algorithms. Examples of the application of the method are given in Section 6. Included are standard, two-dimensional hydrodynamic and MHD shock problems, as well as applications of the method to the equilibrium and stability of toroidal fusion plasmas in two and three dimensions. Our initial results with mesh adaption are also described. The conclusions, summary, and future plans are given in Section 7.

2.0 COMPUTATIONAL MESHES

The computational description of a continuous, time dependent system, such as a magnetized plasma, has three components: a continuum model of the system that describes the evolution of infinitesimally small volume elements for infinitesimally small intervals of time; an approximation to the continuum model that describes the evolution of finite sized volume elements for infinitesimally small intervals of time; and, a description of how these finite sized volume elements evolve over finite time intervals. In this work we have chosen resistive magnetohydrodynamics as the continuum model. This will be described in Section 3. The finite temporal description will be given in Section 5. Here, and in Section 4, we will discuss finite methods of spatial representation.

2.1 Finite-dimensional Systems and Structured Meshes.

Continuous systems described by partial differential equations respond to differences between the state of the system at one spatial location and the state at another spatial location that is only infinitesimally distant. The state of the system is defined on a continuum of points in the domain. In a finite analog of such a system, the infinity of points in the continuum is replaced by a finite number of discrete points, and infinitesimal distance is replaced by the finite distance between neighboring points. For the purposes of computing the differences in the state of the system between these points, near neighboring points can be thought of as being linked together to form a mesh that covers the domain. The description of the mesh consists of a list of the mesh points and their connectivity. The physical relationships between the state of the system at one mesh point and that at all others then defines a finite-dimensional set of nonlinear algebraic equations that are the exact equations of motion for the finite system. The extent to which the dynamics of this finite dimensional system approximate those of the continuum system determines the accuracy and utility of the approximation.

A *structured* mesh is one in which a pre-defined logical structure (or order) is assumed to exist. For example, in 2D Cartesian coordinates, a structured mesh consists of a product of two sets of mesh arrays (the x and y coordinates), with indices i and j, ordered by increasing coordinate value. Two indices are required to identify a mesh point: point (i, j) has coordinates $x(i)$, $y(j)$. The mesh is structured logically so that points $(i + 1, j)$ and $(i, j + 1)$ are adjacent to point (i, j). This logical structure is assumed to hold for all points in the domain, and is implicitly used in constructing the finite-dimensional algebraic equations that describe the dynamical evolution of the finite system. Structured meshes form the familiar quadrilateral grids commonly used in numerical methods. The boundary of the domain naturally consists of curves of the form $x = constant$ and $y = constant$. (An irregular domain would be built up from unions of such meshes.) As neighboring points are logically connected in this way, adding and deleting points affects the indexing of *all* points in the mesh.

2.2 Unstructured Meshes

In contrast to a structured mesh, an *unstructured* mesh is one that has no pre-defined logical structure. An unstructured mesh consists of a set of arbitrarily ordered points. A single mesh index suffices to identify a point. Point i , having coordinates x_i and y_i, and point $i + 1$, having coordinates x_{i+1} and y_{i+1}, are not necessarily adjacent.

Since there is no pre-defined logical structure, the mesh points are not constrained to lie along any predetermined curves; they may be arbitrarily distributed in the domain. Neighboring points are then connected by line elements to form a mesh of triangles that covers the domain. The mesh points $\mathbf{r}_i$ form the vertices of the triangles, and the connecting lines form the triangle edges. (This triangulation is not unique. This will be discussed in Section 2.4.) The mesh consists of N_v vertices, N_e edges, and N_s triangles, with $N_v < N_s < N_e$.

With each triangle s we will associate a point $\mathbf{r}_s$. This point identifies the location of the triangle in the domain. (Like the triangulation, the definition of $\mathbf{r}_s$ is not unique. This will be discussed in Sections 2.4 and 2.5.) It is also convenient to define the edges of the triangles as directed line segments, or vectors $\mathbf{l}_{i,j}$, connecting point i with point j, i.e., $\mathbf{l}_{i,j} = \mathbf{r}_j - \mathbf{r}_i$. Every edge e thus has triangle $s = L$ on the left, and triangle $s = R$ on the right. With each edge we also associate a unit tangent vector $\mathbf{t}_e = \mathbf{l}_e / l_e$, and a unit normal vector $\mathbf{n}_e$ that points from the left side to the right side. These mesh elements are sketched in Figure 2-1.

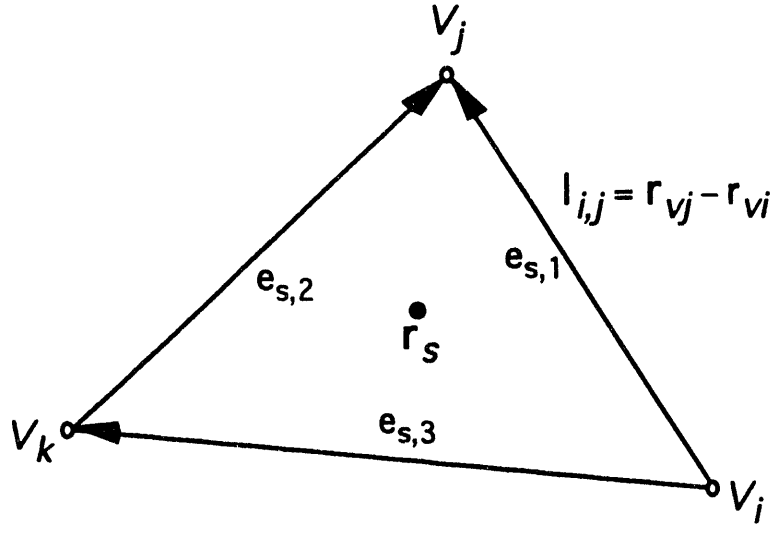


Figure 2-1. Triangle, edge, and vertex mesh elements.

An unstructured mesh is identified and manipulated by means of primary and secondary data sets. The primary data set consists of a list of mesh elements. Secondary data sets define the connectivity between the primary mesh elements. For example, for 2D meshes the spatial representation consists of triangular elements. The primary data set consists of a list of cells (triangles), their vertices, and the edges connecting them. Additional data sets consist of cross-indexing information that relate the elements of the primary set. For example, an edge-indexed array specifies the indices of the cells to the left and right of an edge. Other cell-indexed arrays specify the indices of the three vertices and three edges of a cell.

Various tools and routines exist for generating and manipulating unstructured triangular meshes. We have adapted several subroutines used in 2-D hydrodynamics codes for use in magnetohydrodynamics. These routines interface with unstructured meshes generated by the SMART code[9]. The SMART code runs interactively on a MACII, and generates files that can be read as input. We have used SMART to generate unstructured meshes for fusion applications. An example of zoning for the poloidal cross-section of the D-III-D experiment is shown in Figure 2-2. Detail of the zoning in the region near the divertor is shown in Figure 2-3.

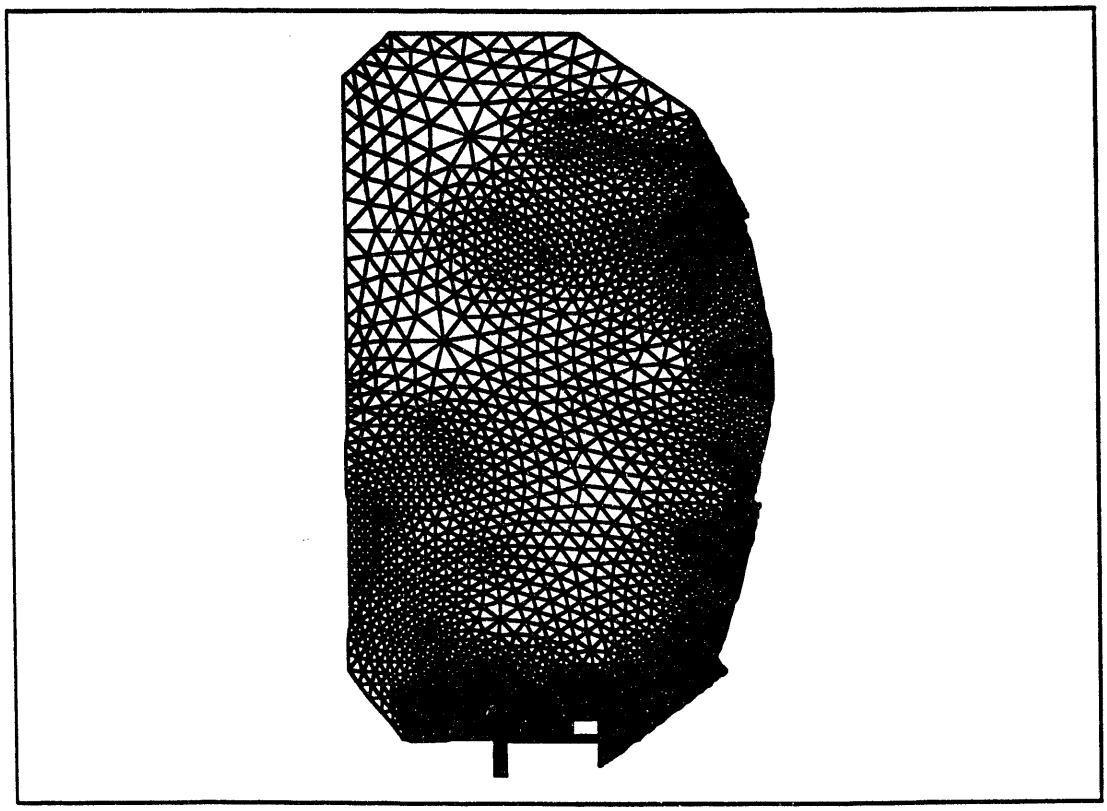


Figure 2-2. An example of an unstructured mesh describing the geometry of the D-III-D experiment.

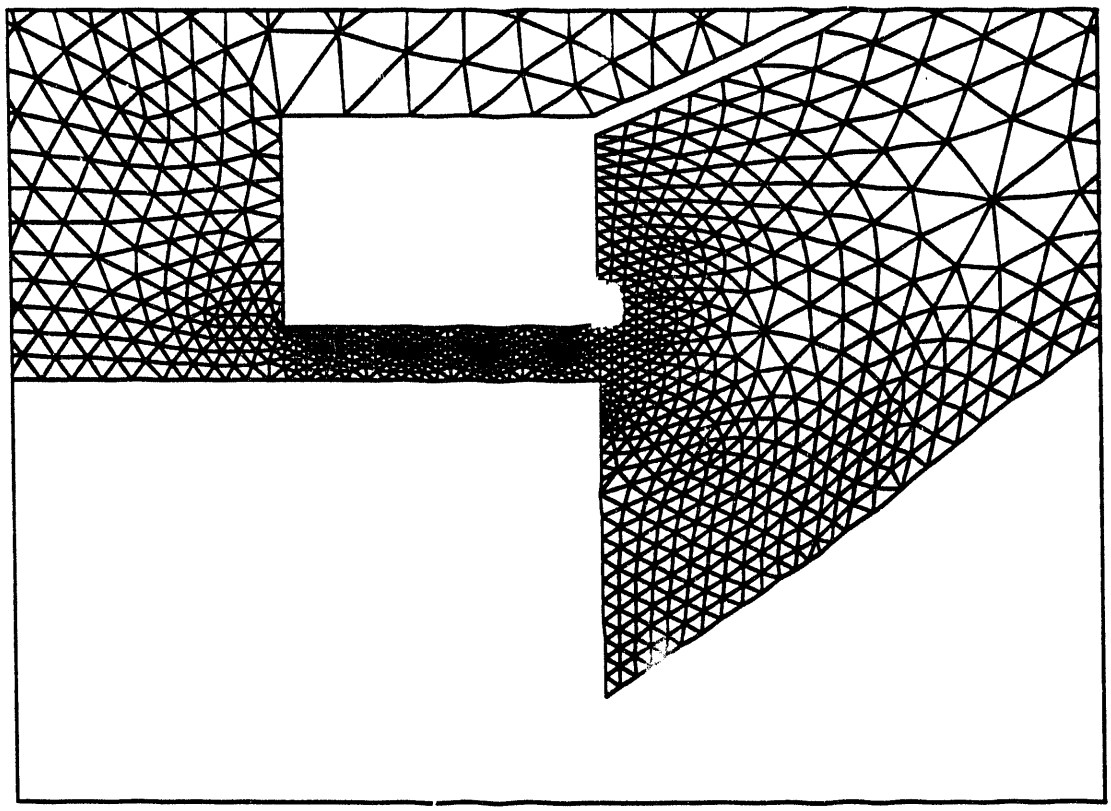


Figure 2-3. Details of the zoning in the divertor region of D-III-D (the bottom portion of Figure 2-2).

2.3 Primary and Dual Meshes

Computational meshes, both structured and unstructured, are used not only to describe geometric regions, but also to define differential operators. For the latter purpose, it is useful to introduce the concept of primary and dual meshes. For a structured mesh, these are often referred to as staggered meshes. Some variables are defined on the primary mesh, and others on the dual, or staggered, mesh. An example of a 2-dimensional staggered mesh is shown in Figure 2-4. These meshes have been used very successfully in MHD simulation[10]

The concept of primary and dual meshes can be extended to triangular meshes. In this case the primary mesh consists of the triangulation of arbitrarily placed points in the plane. The mesh points are the vertices of the triangles. The dual mesh consists of polygons that surround each vertex. The vertices of the dual polygons can be chosen in several ways. Two choices will be discussed below. When taken together, the primary triangular mesh and the dual polygon mesh are the generalization of structured, staggered meshes. An example of a triangular mesh and its polygon dual are sketched in Figure 2-5.

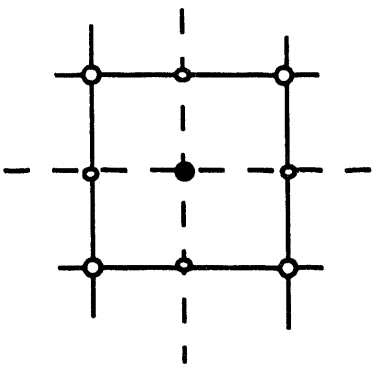


Figure 2-4. Structured, staggered (dual) meshes.

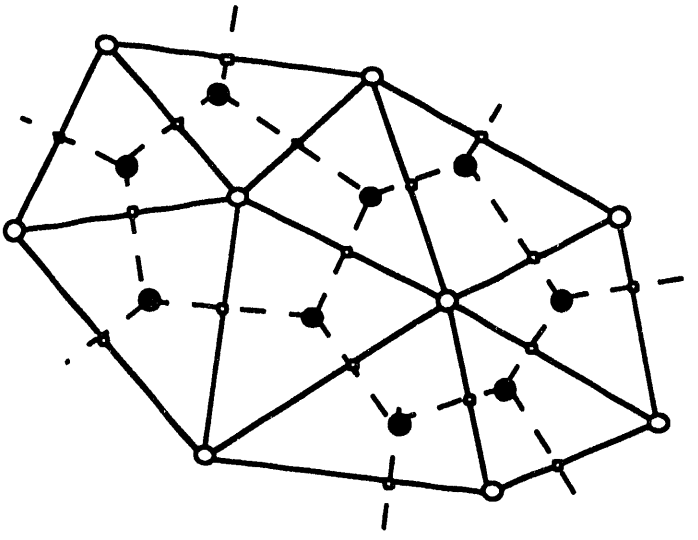


Figure 2-5. Triangular (primary) and polygon (dual) meshes.

2.4 Delaunay Triangles and Voronoi Polygons

A set of points in a plane can be connected to form triangular cells that cover the plane. The connectivity of the resulting mesh is not unique, i.e., there are many ways to connect a given set of points. One triangulation that has several desirable properties is the *Delaunay triangulation*. This is described below.

Consider a set of points P arbitrarily distributed in the plane. A *Voronoi polygon* is defined as the boundary of the region surrounding a point P_i within which all points in the plane are closer to P_i than to any other member of the set P. The mesh consisting of all the Voronoi polygons of the set P is called the Voronoi (or Dirichlet) tessellation of the plane. The Delaunay triangles form a mesh that is dual to the Voronoi polygons. It is formed by connecting points whose Voronoi polygons share a common side.

The Voronoi/Delaunay mesh has several interesting and desirable properties. Some of these[11] are stated here without proof.

1. A Delaunay triangulation of a set P is unique.
2. The vertices of the Voronoi polygons (these points are *not* members of the set P) are the circumcenters of the Delaunay triangles (i.e., a Voronoi vertex is the center of a circle passing through the three vertices of a Delaunay triangle.)
3. The sides of the Voronoi polygons are perpendicular bisectors of the sides of the Delaunay triangles. (These dual meshes are orthogonal.)
4. The Delaunay triangulation maximizes the minimum angle of the triangulation; i.e., of all triangulations of the set P the Delaunay triangles are the closest to being equiangular, on average. A Delaunay triangulation is said to be acute if the interior angles of all triangles are acute. (In general, the Delaunay triangulation of an arbitrarily distributed set P is *not* acute.)
5. An edge formed by joining a point P_i to its nearest neighbor is an edge of a Delaunay triangle.
6. In an acute Delaunay triangulation, the vertices of the Voronoi polygons are always interior to their corresponding Delaunay triangles.

As we will discuss in Section 4, Properties 3, 5, and 6 make acute Delaunay triangles very desirable for defining discrete approximations to differential operators. Several algorithms exist for constructing the Delaunay triangulation of the set of points P.

Unfortunately, acute triangulations are not guaranteed for arbitrarily distributed points. When the triangulation is not acute, the Voronoi vertices are no longer interior to their corresponding Delaunay triangles. This makes the use of this dual mesh undesirable. In the future, we are planning to study ways of distributing, or redistributing, the points P to guarantee an acute triangulation. For the present, we use a slightly different dual mesh.

2.5 The Barycenter, or Centroid, Dual Mesh

Because of Properties 4 and 5 of Section 2.4 we use a Delaunay triangulation for the primary mesh. However, instead of Voronoi polygons we use a dual mesh whose vertices are the centroids, or barycenters, of each triangle. If the coordinates of the triangle vertices (the points P) are $\mathbf{r}_v$, the coordinates of the vertices of the dual mesh are given by

$$\mathbf{r}_s = \frac{1}{3}\left(\mathbf{r}_{v1} + \mathbf{r}_{v2} + \mathbf{r}_{v3}\right), \; s = 1, 2, \ldots, N_s \tag{2.1}$$

where N_s is the number of triangles and the $\mathbf{r}_{vi}$ are the three vertices of triangle s. This dual mesh has the property that the vertices of the polygons are always interior to their corresponding triangles. It has the undesirable property that the edges of the dual meshes are no longer orthogonal. This complicates the definition of differential operators, as will be discussed in Section 4.

2.6 The Third (Toroidal) Dimension

Our goal is to describe magnetohydrodynamics in toroidal fusion systems. We use the unstructured Delaunay triangular mesh and the barycenter dual mesh dual to approximate the geometry in the poloidal (r, z) plane. It remains to approximate the angular toroidal (ϕ) dimension. Since this coordinate direction is periodic, and since the dominant MHD motions in a tokamak are long wavelength, we have chosen a pseudospectral description using Fast Fourier Transforms (FFTs) for this dimension. The toroidal mesh is thus structured, with a uniform mesh spacing $\Delta\phi = 2\pi/N_\phi$, where N_ϕ is the number of toroidal mesh points; N_ϕ must be a power of 2.

The three-dimensional control volume is sketched in Figure 2-6. The elemental volume is $\Delta V_s = r_s \Delta\phi \Delta a_s$, where Δa_s is the planar area of triangle s and r_s is the radius of the triangle centroid. The Pappus-Guldinus Theorem guarantees that this formula is exact.

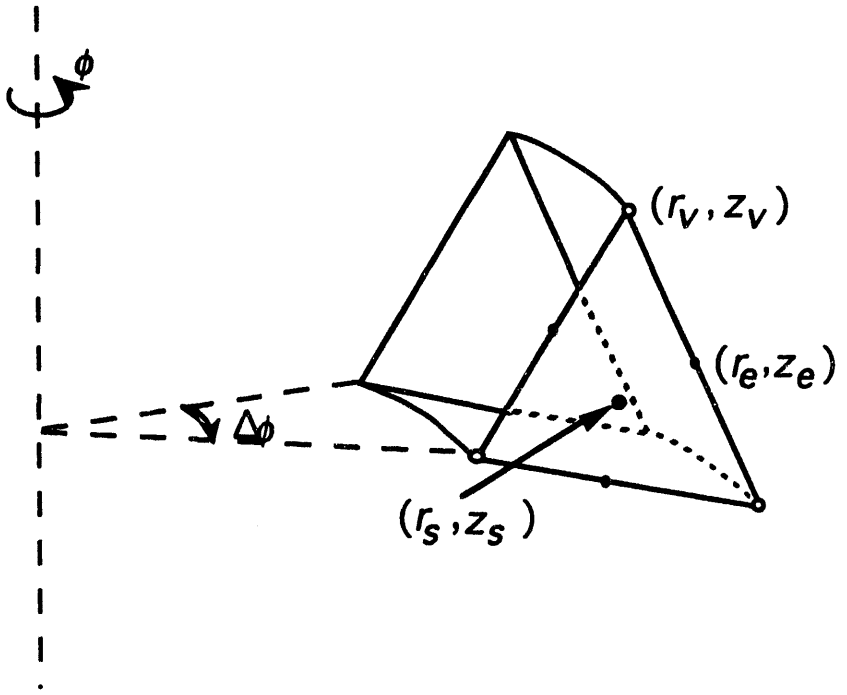


Figure 2-6. Three-dimensional control volume.

2.7 Mesh Refinement

The use of an unstructured mesh allows for new triangles to be added, and old ones deleted, in a relatively easy manner. New triangles are merely added to the end of the list, and old triangles deleted and the list shortened. A new triangle is added by introducing a new vertex at the centroid of a triangle to be refined. New edges connect this vertex to the three vertices of the original triangle. The original triangle is thus divided into three, and two new triangles, three new edges, and one vertex are added to the lists. The new edges may need to be swapped between the new vertex and the opposing vertices of the three neighboring triangles. The circumcenter test[12] is used to determine whether or not edge swapping is required. The new triangulation is thus as acute as possible. The addition of a vertex and edge swapping are sketched in Figure 2-7. The triangle centered densities can then be distributed over the new triangles in a conservative manner.

Before adaption can occur a triangle must be identified for refinement. This can be done by using the edge fluxes to estimate the change in cell quantities during the next time step, or by estimating the gradient of cell quantities in neighboring triangles. These methods have been successful in hydrodynamics. The most appropriate method for MHD is still to be identified, and is a subject of active investigation.

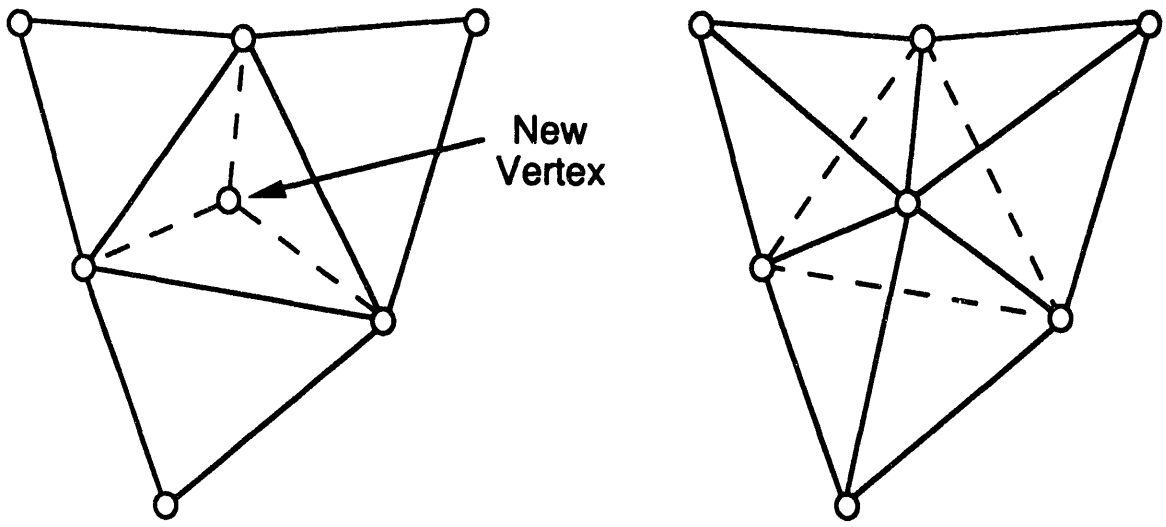


Figure 2-7. Mesh refinement and edge swapping.

3.0 APPROXIMATION TO SPATIAL DIFFERENTIAL OPERATORS

We now proceed to define approximations on the triangular, unstructured mesh to the differential operators that appear in the MHD equations. We use the method of finite volumes as applied to the three-dimensional volume element shown in Figure 2-6.

Consider the triangle in the poloidal (r, z) plane shown in Figure 3-1. We define normal and tangent unit vectors $\mathbf{n}_e$ and $\mathbf{t}_e$ at each edge such that

$$\mathbf{e}_\phi = \mathbf{t}_e \times \mathbf{n}_e \tag{3.1}$$

where $\mathbf{e}_\phi$ is the toroidal unit vector. (Note that $\mathbf{e}_\phi$ points "into" the page.) The normal and tangent unit vectors are given by

$$\mathbf{l}_e = \Delta r_e \, \mathbf{e}_r + \Delta z_e \, \mathbf{e}_z = \Delta l_e \mathbf{t}_e \tag{3.2}$$

$$\mathbf{n}_e = \frac{\Delta z_e \, \mathbf{e}_r - \Delta r_e \, \mathbf{e}_z}{\Delta l_e} \tag{3.3}$$

and the area of the triangle is

$$\Delta a_s = \frac{1}{2} |\mathbf{l}_1 \times \mathbf{l}_2| = \frac{1}{2} |\mathbf{l}_2 \times \mathbf{l}_3| = \frac{1}{2} |\mathbf{l}_3 \times \mathbf{l}_1| \, . \tag{3.4}$$

We say that $\mathbf{n}_e$ points from the left side of edge e (triangle $s = L_e$) to the right side of edge e (triangle $s = R_e$).

We use a finite volume method to obtain the approximations to the differential operators. In this method differential operators are defined in terms of their integral relations. We assume all functions are of the form

$$f(r,\phi,z,t) = \sum f_n(r,z,t) \, e^{in\phi} \tag{3.5}$$

and then integrate the appropriate identity over the three-dimensional control volume shown in Figure 2-6. This technique assures that the same integral relationships are obeyed by the finite difference approximations and their equivalent differential operators.

To obtain an approximation for the *gradient of a scalar* we substitute Equation (3.5) into the integral identity

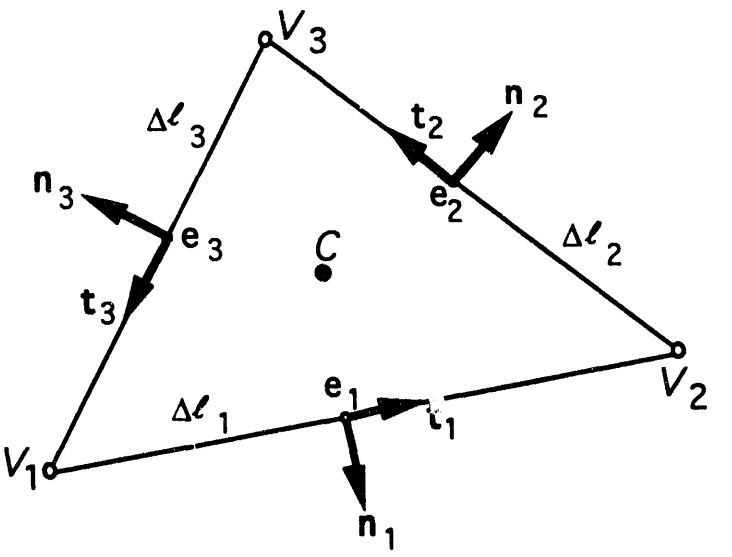


Figure 3-1. Poloidal projection of control volume.

$$\int \nabla f dV = \oint f \mathbf{n} dS \,, \tag{3.6}$$

and use second order approximations to the volume and surface integrals to obtain

$$(\nabla f)_s = \frac{1}{r_s \Delta a_s} \sum r_e \Delta l_e \mathbf{n}_e \bar{f}_e - \frac{f_s}{r_s} \mathbf{e}_r + \frac{in}{r_s} f_s \mathbf{e}_\phi \,. \tag{3.7}$$

The sum is taken over the three edges of triangle s, and the radius of edge e is $r_e = (r_{ve1} + r_{ve2})/2$, where r_{ve1} and r_{ve2} are the radial coordinates of the vertices connected by edge e.

Similarly, for the *divergence of a vector* we use the identity

$$\int \nabla \cdot \mathbf{A} dV = \oint f \mathbf{n} \cdot \mathbf{A} dS \tag{3.8}$$

to obtain the approximation

$$(\nabla \cdot \mathbf{A})_s = \frac{1}{r_s \Delta a_s} \sum r_e \Delta l_e \mathbf{n}_e \cdot \mathbf{A}_e + \frac{in}{r_s} A_{\phi s} \,; \tag{3.9}$$

for the *curl of a vector* we use

$$\int \nabla \times \mathbf{A} dS = \oint \mathbf{t} \cdot \mathbf{A} dl \tag{3.10}$$

to obtain the approximation

$$(\nabla \times \mathbf{A})_{ne} = -\frac{1}{r_e \Delta l_e} \left(r_{v+} A_{\phi v+} - r_{v-} A_{\phi v-} \right) + \frac{in}{r_e} \mathbf{A}_e \cdot \mathbf{t}_e \,, \tag{3.11}$$

$$(\nabla \times \mathbf{A})_{\phi s} = -\frac{1}{\Delta a_s} \sum \mathbf{A}_e \cdot \mathbf{l}_e \,. \tag{3.12}$$

Here we have taken surface and line integrals over the faces of the control volume and their respective bounding edges. To approximate the *divergence of a tensor* we use

$$\int \nabla \cdot \mathbf{T} dV = \oint f\mathbf{n} \cdot \mathbf{T} dS \tag{3.13}$$

to obtain

$$\begin{aligned} (\nabla \cdot \mathbf{T})_s = {} & \frac{1}{r_s \Delta a_s} \sum r_e \Delta l_e \left[\mathbf{e}_r (\mathbf{n}_e \cdot \mathbf{T}_e)_r + \mathbf{e}_\phi (\mathbf{n}_e \cdot \mathbf{T}_e)_\phi + \mathbf{e}_z (\mathbf{n}_e \cdot \mathbf{T}_e)_z \right] \\ & + \frac{in}{r_s} \left[\mathbf{e}_r T_{\phi rs} + \mathbf{e}_\phi T_{\phi \phi s} + e_z T_{\phi z s} \right] + \frac{1}{r_s} \left[e_\phi T_{\phi rs} - e_r T_{\phi \phi s} \right] . \end{aligned} \tag{3.14}$$

It is easy to verify from Equations (3.9), (3.11), and (3.12) that $\nabla \cdot \nabla \times \mathbf{A} \equiv 0$ for these finite operators. Solenoidal magnetic fields are thus assured. This is a direct result of the use of consistent integral relations to obtain the finite approximations.

4.0 THE MHD EQUATIONS:

PLACEMENT OF THE VARIABLES ON THE MESH

In this work we solve the equations of resistive MHD. In a convenient nondimensional form, they are

$$\frac{\partial \mathbf{A}}{\partial t} = - \mathbf{E} \tag{4.1}$$

$$\mathbf{E} = -\mathbf{v}\times\mathbf{B}+\eta\mathbf{J}/S \tag{4.2}$$

$$\mathbf{B} = \nabla\times\mathbf{A} \tag{4.3}$$

$$\mathbf{J} = \nabla\times\mathbf{B} \tag{4.4}$$

$$\frac{\partial \rho\mathbf{v}}{\partial t} = - \nabla\cdot\mathbf{T} \tag{4.5}$$

$$\mathbf{T} = \rho\mathbf{v}\mathbf{v}-\mathbf{B}\mathbf{B} + \frac{1}{2}\left(p+B^2\right)\mathbf{I} \tag{4.6}$$

$$\frac{\partial \rho}{\partial t} = - \nabla\cdot(\rho\mathbf{v}) \tag{4.7}$$

$$\frac{\partial u}{\partial t} = - \nabla\cdot\mathbf{F} \tag{4.8}$$

$$u = \rho v^2+B^2 + \frac{p}{\gamma-1} \tag{4.9}$$

$$\mathbf{F} = \left(\rho v^2+\frac{\gamma}{\gamma-1}\,p\right)\mathbf{v}+2\ \mathbf{E}\times\mathbf{B} \tag{4.10}$$

where γ is the ratio of specific heats, u is the total energy density, $\mathbf{F}$ is the energy flux, $\mathbf{I}$ is the unit tensor, and $\mathbf{T}$ is the Reynolds-Maxwell stress tensor. All other

quantities have their usual meanings. Following the experience of hydrodynamics, we have chosen the conservation form of the equations.

4.1 Hydrodynamic Variables

The boundary of the poloidal computational region is formed by triangle edges. As is the practice in computational fluid dynamics, we define the momentum density $\rho\mathbf{v}$, the energy u and the mass density ρ at the triangle centroids $\mathbf{r}_s$. These quantities thus represent the momentum, energy, and mass per unit volume in a triangular cell. Velocities in a cell are given by $\mathbf{v}_s = (\rho\mathbf{v})_s/\rho_s$. The rate of change of these quantities given by applying the differential approximations defined in Section 3.2 to Equations (4.1-4.10). For example, the rate of change of energy density in triangle s is given by

$$\frac{\partial u_s}{\partial t} = -\frac{1}{r_s \Delta a_s} \sum r_e \Delta l_e \mathbf{n}_e \cdot \mathbf{F}_e + \frac{in}{r_s} F_{\phi s} \tag{4.11}$$

where

$$\mathbf{n}_e \cdot \mathbf{F}_e = \left(\rho v^2 + \frac{\gamma}{\gamma - 1} p\right)_e v_{ne} + 2\left(\overline{E}_{\phi e} B_{te} - E_{te}\overline{B}_{\phi e}\right) \tag{4.12}$$

is the poloidal energy flux across edge e, and

$$F_{\phi s} - \left(\rho v^2 + \frac{\gamma}{\gamma - 1} p\right)_s v_{\phi s} + 2\left(\overline{E}_{ts} B_{ns} - E_{ns} B_{ts}\right) \tag{4.13}$$

is the energy flux in the toroidal direction. The quantity v_{ne} is the normal component of velocity at edge e, and is defined as

$$v_{ne} = \frac{1}{2} \mathbf{n}_e \cdot (\mathbf{v}_{Le} + \mathbf{v}_{Re}) \tag{4.14}$$

Expressions similar to Equations (4.11-4.13) hold for the momentum equation (Eq. 4.5), and the continuity equation (Eq. 4.7).

The advective flux at an edge e is computed using the full donor cell method. For example, in the first term on the right hand side of Equation (4.12) is evaluated as

$$(F_{ne})_{adv} = \left(\rho v^2 + \frac{\gamma}{\gamma - 1}\, p\right)_{Le} v_{ne} \ , \quad \text{if} \quad v_{ne} > 0 \tag{4.15a}$$

$$(F_{ne})_{adv} = \left(\rho v^2 + \frac{\gamma}{\gamma - 1}\, p\right)_{Re} v_{ne} \ , \quad \text{if} \quad v_{ne} < 0 \tag{4.15b}$$

where L_e and R_e are the indices of the triangles to the left and right of edge e, respectively. This method introduces numerical diffusion of order $v_{ne}\Delta/2$, where $\Delta \approx \Delta a_s{}^{1/2}$. While this technique may be too diffusive for shock calculations, it is quite adequate to describe the relatively slow motions of interest in tokamak dynamics. Problems involving strong shocks will require a higher order treatment.

4.2 Electromagnetic Variables

The primary electromagnetic variable in this formulation is the vector potential **A**. We define A_r and A_z at the triangles edges e, and A_ϕ at the triangle centroids s. Then Equations (3.11) and (3.12) define B_{ne}, the component of **B** in the poloidal plane normal to a triangle edge, and $B_{\phi s}$ the toroidal component of **B** at the triangle centroid. (Note that $(\nabla \cdot \mathbf{B})_s \equiv 0$.)

To uniquely determine the magnetic field we must also define another independent component of **B** in the poloidal plane. This is done by integrating Equation (3.10) over the surface of the dual polygon p_e that crosses an edge e, as shown in Figure 4-1. The polygon edge has a unit normal vector $\mathbf{n}_{pe}$ and a unit tangent vector $\mathbf{t}_{pe}$. This defines B_{npe}, the component of **B** normal to the polygon edge. The cylindrical components of the poloidal field are then given by

$$B_{re} = \frac{1}{\Delta}\left(B_{ne} n_{pez} - B_{npe} n_{ez}\right) \tag{4.16}$$

$$B_{ze} = \frac{1}{\Delta}\left(B_{npe} n_{er} - B_{ne} n_{per}\right) \tag{4.17}$$

where

$$\Delta = e_\phi \cdot \left(\mathbf{n}_e \times \mathbf{n}_{pe}\right) \neq 0 \ , \tag{4.18}$$

from which the tangential component of **B** at edge e is computed as

$$B_{te} = B_{ze} n_{er} - B_{re} n_{ez} \ . \tag{4.19}$$

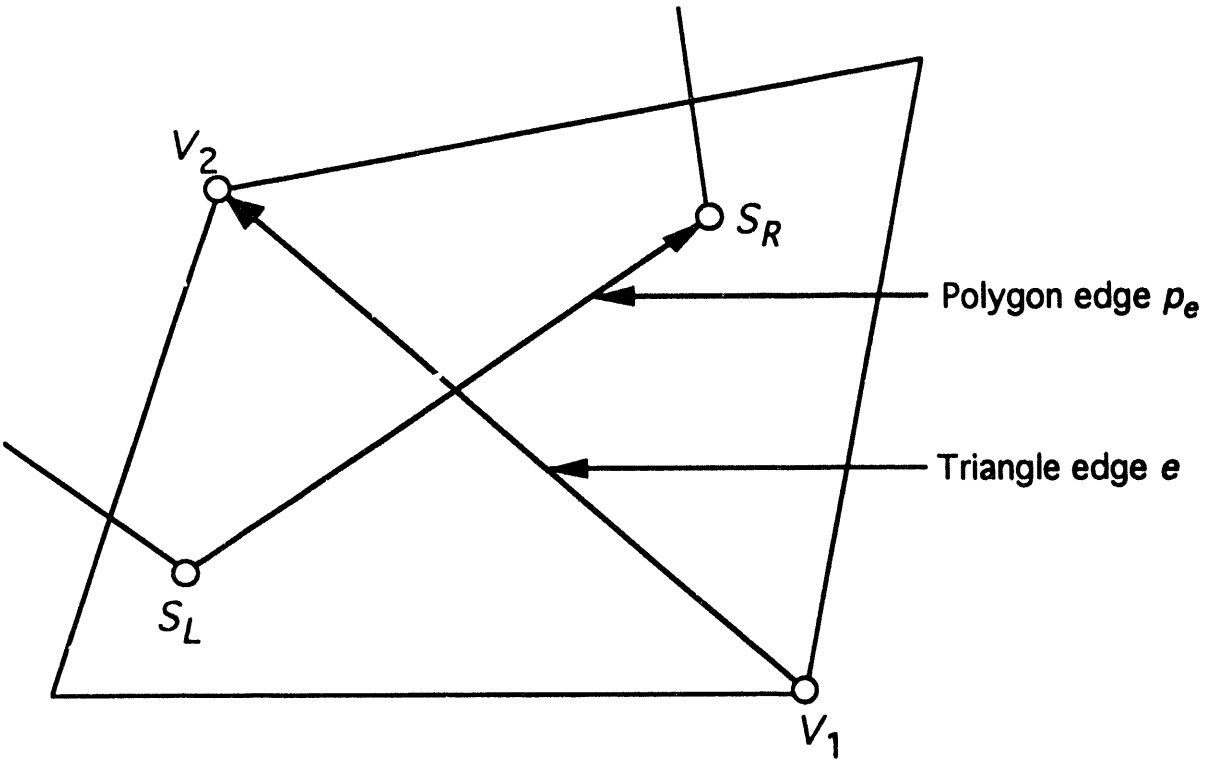


Figure 4-1. Triangle and polygon edges.

Similar relationships hold for the current density **J**. (Note that if the mesh consists of Delaunay triangles and Voronoi polygons the dual meshes are orthogonal and this calculation is simplified.)

In light of Equation (4.1), we define the electric field **E** at the same spatial locations as the vector potential **A**. The normal and tangential components of the electric field at a triangle edge are given by

$$E_{ne} = - \bar{v}_{\phi e} B_{te} + \bar{v}_{ne} \bar{B}_{te} + \eta J_{ne} / S , \tag{4.20}$$

$$E_{te} = - \bar{v}_{ne} \bar{B}_{\phi e} + \bar{v}_{\phi e} B_{ne} + \eta J_{te} / S . \tag{4.21}$$

The toroidal electric field at the triangle centroids is given by

$$E_{\phi s} = - v_{zs} \bar{B}_{rs} + v_{rs} \bar{B}_{zs} + \eta J_{\phi s} / S . \tag{4.22}$$

4.3 Averages and Interpolation

In the above formulas, an overscore indicates that an average should be taken, or that interpolation be performed. Several types of interpolation are required in the present algorithm. These are discussed in this section.

Interpolation from triangle centroids to edges is a simple average between adjacent triangles:

$$\bar{f}_e = \frac{1}{2}\left(f_{Re} + f_{Le}\right) . \tag{4.23}$$

Interpolation from vertices or edges to triangle centroids is also a simple average. For functions defined on vertices

$$\bar{f}_s = \frac{1}{3}\left(f_{v1} + f_{v2} + f_{v3}\right) , \tag{4.24}$$

and for functions defined at edges

$$\bar{f}_s = \frac{1}{3}\left(f_{e1} + f_{e2} + f_{e3}\right) , \tag{4.25}$$

where $v1$, $v2$, $v3$, and $e1$, $e2$, $e3$ are the three vertices and edges of triangle s, respectively.

For interpolation from triangle centroids to vertices, we use a pseudo-Laplacian weighted average[13]. In this approach, the interpolated value of a function at vertex v is given by the weighted average

$$\bar{f}_v = \frac{\sum(1+w_s)f_s}{\sum(1+w_s)} , \tag{4.26}$$

where the sums are taken over all triangles sharing vertex v.

We require that the weights w_s be as small as possible, and that the interpolation be exact for linear functions. We can then determine the w_s by minimizing the functional

$$F(w_s) = \sum w_s^2 \tag{4.27}$$

subject to the constraints

$$L_r = \sum (1+w_s)(r_s - r_v) = 0 , \tag{4.28a}$$

$$L_z = \sum (1+w_s)(z_s - z_v) = 0 , \tag{4.28b}$$

where (r_v, z_v) and (r_s, z_s) are the coordinates of the vertex and the centroids. The result is that

$$w_s = \lambda_r \Delta r_s + \lambda_z \Delta z_s , \tag{4.29}$$

where $\Delta r_s = r_s - r_v$, $\Delta z_s = z_s - z_v$, and λ_r and λ_z are Lagrange multipliers given by

$$\lambda_r = \frac{R_z I_{rz} - R_r I_{zz}}{I_{rr} I_{zz} - I_{rz}^2} , \tag{4.30}$$

$$\lambda_z = \frac{R_z I_{rz} - R_z I_{rr}}{I_{rr} I_{zz} - I_{rz}^2} , \tag{4.31}$$

$$R_r = \sum \Delta r_s , \tag{4.32}$$

$$R_z = \sum \Delta z_s , \tag{4.33}$$

$$I_{rr} = \sum (\Delta r_s)^2 , \tag{4.34}$$

$$I_{zz} = \sum (\Delta z_s)^2 , \tag{4.35}$$

$$I_{rz} = \sum \Delta r_s \Delta z_s . \tag{4.36}$$

Equations (4.23-4.26) are first order accurate, i.e., they are exact for linear functions. (Equations (4.24) and (4.25) are just special cases of Equation (4.26)). The use of higher order interpolation methods, especially in place of Equation (4.23), can be shown to lead to a non-Hermitian formulation and resulting unphysical behavior.

A further complication is that neither Equations (4.23) and (4.25), nor Equations (4.24) and (4.26), are exact inverses of each other. Thus, for example,

interpolation from centroids to vertices using Equation (4.26), followed directly by interpolation from vertices to centroids using Equation (4.24), introduces errors. This is especially important during mesh refinement. We have not yet found a satisfactory solution to this problem . Interpolation on unstructured meshes is an area to be investigated in future work.

4.4 Boundary Conditions

Since the computational boundary consists of triangle edges, the mass density in triangles adjacent to the boundary is completely determined by the normal velocity v_{ne} at the boundary. The momentum density and energy density also require that the normal component of the magnetic field, B_{ne}, and the total pressure, $p + B^2$, be specified. For a non-porous, perfectly conducting wall, the appropriate boundary conditions are $B_{ne} = v_{ne} = 0$, and $(p + B^2)_e = (p + B^2)_s$ (no normal pressure gradient.) Implementation of boundary conditions is aided by introducing ghost triangles that lie outside the boundary and are reflections of interior triangles that contain a boundary edge. For the electromagnetic variables it is sufficient to specify the electric field tangent to the boundary. Thus, for a perfectly conducting wall, $E_{te} = E_{\phi e} = 0$, where $E_{\phi e}$ is the average of the toroidal electric field in a boundary cell and its reflected ghost cell. Note that A_{ne}, and hence E_{ne}, is not required to determine **B** in a boundary triangle.

5.0 TIME INTEGRATION

As is appropriate for sound and Alfvén waves, the time integration algorithm uses an explicit leapfrog method with predictor-corrector steps to stabilize the nonlinear advective terms. The velocity and momentum are defined at time t^n. The energy density, mass density and vector potential are defined at time $t^{n+1/2}$. The time step can be arbitrarily large; the semi-implicit method[8] is used to remove the CFL time-step restriction. Artificial viscosity is treated fully implicitly.

The time advance proceeds by means of operator splitting, i.e.,

$$\left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\text{total}} = \left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\text{explicit}} + \left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\text{semi-implicit}} + \left.\frac{\partial \mathbf{U}}{\partial t}\right|_{\text{viscous}} \tag{5.1}$$

or,

$$\frac{\mathbf{U}^* - \mathbf{U}^n}{\Delta t} = \mathbf{F}_{\text{explicit}} \tag{5.2a}$$

$$\frac{\mathbf{U}^{**} - \mathbf{U}^*}{\Delta t} = \mathbf{F}_{\text{semi-explicit}} \tag{5.2b}$$

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^{**}}{\Delta t} = \mathbf{F}_{\text{viscous}} \tag{5.2c}$$

where U is the state-vector describing the system, and $\mathbf{F}_{\text{explicit}}$, $\mathbf{F}_{\text{semi-implicit}}$, and $\mathbf{F}_{\text{viscous}}$ represent the explicit (wave-like and advective), semi-implicit, and viscous terms that appear on the right-hand side of the equations. Details of these methods are given in the following sections.

5.1 Explicit Advance

Wave-like and advective terms are advanced explicitly with Δt chosen for accuracy and computational convenience rather than numerical stability. The explicit part of the algorithm is:

$$\frac{\mathbf{A}^* - \mathbf{A}^{n-1/2}}{\Delta t} = -\mathbf{v}^n \times \mathbf{B}^{n-1/2} + \eta \mathbf{J}^{n-1/2}/S \tag{5.3}$$

$$\mathbf{B}^* = \nabla \times \mathbf{A}^* \tag{5.4}$$

$$\mathbf{J}^* = \nabla \times \mathbf{B}^* \tag{5.5}$$

$$\frac{\mathbf{A}^{n+1/2} - \mathbf{A}^{n-1/2}}{\Delta t} = - v^n \times \mathbf{B}^* + \eta \mathbf{J}^* / S \tag{5.6}$$

$$\frac{u^* - u^{n-1/2}}{\Delta t} = - \nabla \cdot \frac{\gamma}{\gamma - 1} p^{n-1/2} \mathbf{v}^n \tag{5.7}$$

$$p^* = (\gamma - 1) \left[u^* - \left(\rho v^2\right)^n - \left(B^2\right)^{n+1/2} \right] \tag{5.8}$$

$$\frac{u^{n+1/2} - u^{n-1/2}}{\Delta t} = - \nabla \cdot \left\{ \left[\left(\rho v^2\right)^n + \frac{\gamma}{\gamma - 1} p^* \right] + 2 \mathbf{E}^{n+1/2} \times \mathbf{B}^{n+1/2} \right\} \tag{5.9}$$

$$\frac{\rho^* - \rho^n}{\Delta t} = - \nabla \cdot \left(\rho^{n-1/2} \mathbf{v}^n \right) \tag{5.10}$$

$$\frac{\rho^{n+1/2} - \rho^n}{\Delta t} = - \nabla \cdot \left(\rho^* \mathbf{v}^n \right) \tag{5.11}$$

$$\frac{\rho \mathbf{v}^* - \rho \mathbf{v}^n}{\Delta t} = - \nabla \cdot (\rho \mathbf{v} \mathbf{v})^n \tag{5.12}$$

$$\frac{\rho \mathbf{v}^{n+1} - \rho \mathbf{v}^n}{\Delta t} = - \nabla \cdot \left[(\rho \mathbf{v} \mathbf{v})^* - (\mathbf{B}\mathbf{B})^{n+1/2} \right] - \frac{1}{2} \nabla \left(p + B^2 \right)^{n+1/2} . \tag{5.13}$$

The predictor-corrector steps introduce an additional diffusion of order $v\Delta t/2$ that can exceed the diffusion from the donor cell fluxes when the time step exceeds the explicit CFL stability limit.

5.2 Semi-Implicit and Implicit Solutions

We use the semi-implicit method[8] to remove the CFL time step restriction for numerical stability associated with the explicit advance described in Section 5.1. This restriction is of the form $C\Delta t/\delta < 1$, where C is the characteristic speed for the propagation of normal modes (Alfvén or sound waves) and δ is a measure of the linear size of a zone (here proportional to the square root of the triangle area). With the use of the semi-implicit method the algorithm becomes numerically stable at arbitrary Δt so that the time step can be chosen for reasons of accuracy or computational convenience rather than numerical stability. This is especially important for tokamak simulations where the time scales set by the normal modes differ by a factor of order $(R/a)^2$. The time step remains limited the advective CFL stability condition $V\Delta t/\delta < 1$, where V is the local flow speed. This is not a significant restriction when $V/C << 1$, as is the case for many fusion applications. When $V/C \approx 1$, as is the case for shocks, the algorithm becomes explicit. This restriction can thus be viewed as an accuracy condition.

In this work we use a simple vector Laplacian semi-implicit operator. This term is added to and subtracted from the right hand side of the momentum equation at the new and old time levels. The semi-implicit advance is

$$\left(1-\alpha\Delta t\nabla^2\right)(\rho\mathbf{v})^{**} = (\rho\mathbf{v})^{*} - \alpha\Delta t(\rho\mathbf{v})^{n} , \tag{5.14}$$

where α is the semi-implicit coefficient given by

$$\alpha = \frac{1}{4}\left(\frac{\Delta t}{\Delta t_{\mathrm{CFL}}} - 1\right) \text{ for } \Delta t > \Delta t_{\mathrm{CFL}} , \tag{5.15}$$

$$\alpha = 0 \text{ for } \Delta t \leq \Delta t_{\mathrm{CFL}} , \tag{5.16}$$

$(\rho\mathbf{v})^{*}$ is the value of momentum obtained from the explicit advance, and Δt_{CFL} is the maximum time step allowed by the CFL restriction for normal modes.

The time advance is completed with the implicit viscous advance

$$\left(1-\nu\Delta t\nabla^2\right)(\rho\mathbf{v})^{n+1} = (\rho\mathbf{v})^{**} , \tag{5.17}$$

where ν is a (possibly spatially dependent) artificial viscosity coefficient.

The vector Laplacian operator appearing in Equations (5.15) and (5.17) require the definition of the scalar Laplacian. This is accomplished by the successive

application of the gradient and divergence operators defined in Equations (3.7) and (3.9). When combined with the boundary condition

$$(\nabla f)_G = (\nabla f)_B \,, \tag{5.18}$$

where the subscripts G and B represent values in ghost and boundary triangles, respectively, the resulting operator is self-adjoint.

Since the mesh is unstructured, the $N_S \times N_S$ matrices corresponding to the operators appearing in Equations (5.14) and (5.17) are not banded but are sparse. An example of the structure pattern for a case with 320 triangles is shown in Figure 5-1. Matrix inversion is performed with a conjugate gradient (CG) algorithm with diagonal preconditioning. Since this method is iterative, the full $N_S \times N_S$ never needs to be stored. Good convergence properties have been found even with relatively large time steps.

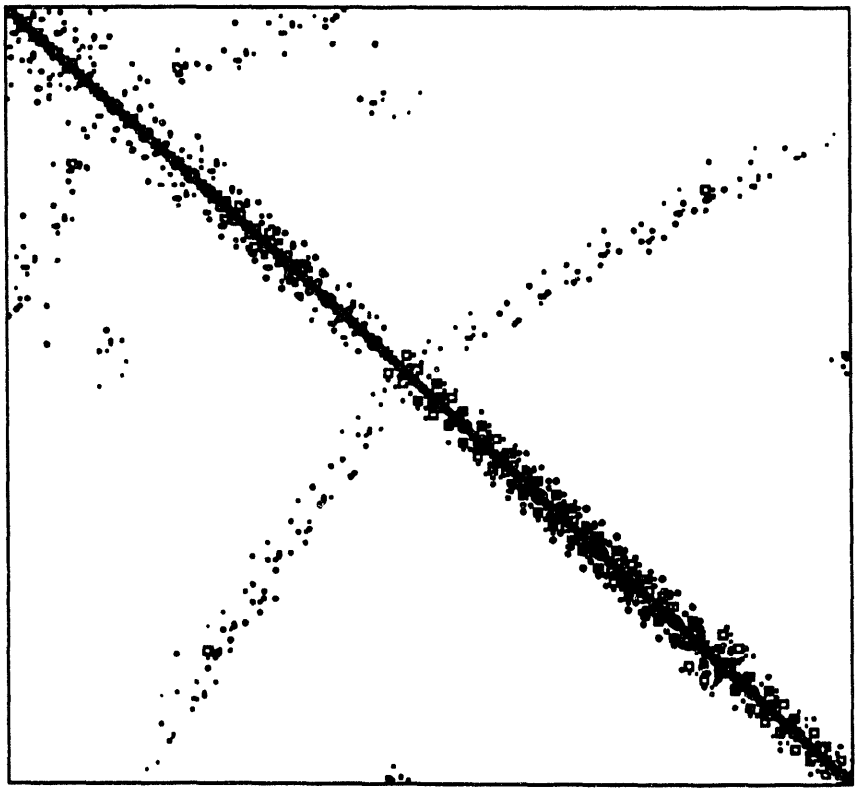


Figure 5-1. Matrix structure for the Laplacian operator on an unstructured mesh.

At this time the resistivity is treated explicitly. Since $\eta/S << 1$ we have not found this to be computationally restrictive. An implicit treatment will require inversion of an operator **L** of the form

$$\mathbf{L}\cdot\mathbf{A} = \mathbf{A} - c\Delta t\nabla\times\nabla\times\mathbf{A} \ . \tag{5.19}$$

We plan to implement this inversion in future versions of the algorithm.

6.0 APPLICATIONS

The algorithm described above has been applied to several nonlinear test problems, both two- and three-dimensional. The code based on the algorithm is called TRIM, for TRIangular MHD. The application of TRIM to these test problems is described in the following sections.

6.1 The Hydrodynamic Shock Tube Problem

A standard problem for testing hydrodynamic algorithms has been defined by Sod[14]. The initial conditions consist of two fluids with different uniform properties separated by a membrane. The fluid to the left of the membrane has pressure p_L and density ρ_L equal to 1. The fluid to the right of the membrane has $p_R = 0.1$, and $\rho_R = 0.125$. The intital velocity is zero and the ratio of specific heats is $\gamma = 1.4$ (air). The magnetic field is zero. These conditions are sketched in Figure 6-1.

At $t = 0$ the membrane is ruptured and the fluid reacts dynamically. This Riemann problem is one of the few fully nonlinear problems that has a known analytic solution[7], and is therefore valuable for testing numerical algorithms. The solution consists of an expansion wave traveling to the left, and a shock wave and a contact discontinuity traveling to the right, all with known velocities.

We have applied the TRIM algorithm to this problem. The time integration is explicit and the artificial viscosity ν is set to zero. While this test problem is one-dimensional, the triangular grid in TRIM requires that a two-dimensional problem be solved. The mesh is shown in Figure 6-2. In this figure, the initial membrane is horizontal. As the solution proceeds in time no spatial variation develops in the direction parallel to the membrane. Contours of this solution are superimposed on the mesh in Figure 6-3. The solution thus remains one-dimensional, even with the two-dimensional algorithm.

A comparison of the results of TRIM with the analytic solution at $t = 0.1$ is shown in Figures 6-4a-c. The two solutions are generally in good agreement. The magnitude of the pressure and velocity in the region between the shock and the expansion fan are quite accurate. (Note the because of the normalization the pressure in TRIM appears to be twice the pressure in the analytic solution.) As is anticipated, the numerical diffusion introduced by the first-order upwind treatment of the interface fluxes has resulted in a considerable smoothing of the discontinuities. This is especially noticeable in the density. The contact discontinuity, which is an interface separating regions of different density but equal pressure and velocity, has been considerably smeared out. This structure is particularly difficult to treat numerically. In contrast with a shock, there are no nonlinear processes that continue to generate a contact discontinuity in opposition to numerical diffusion; it is merely an interface between two states of different density. The effect of any diffusion in the algorithm is felt most strongly here.

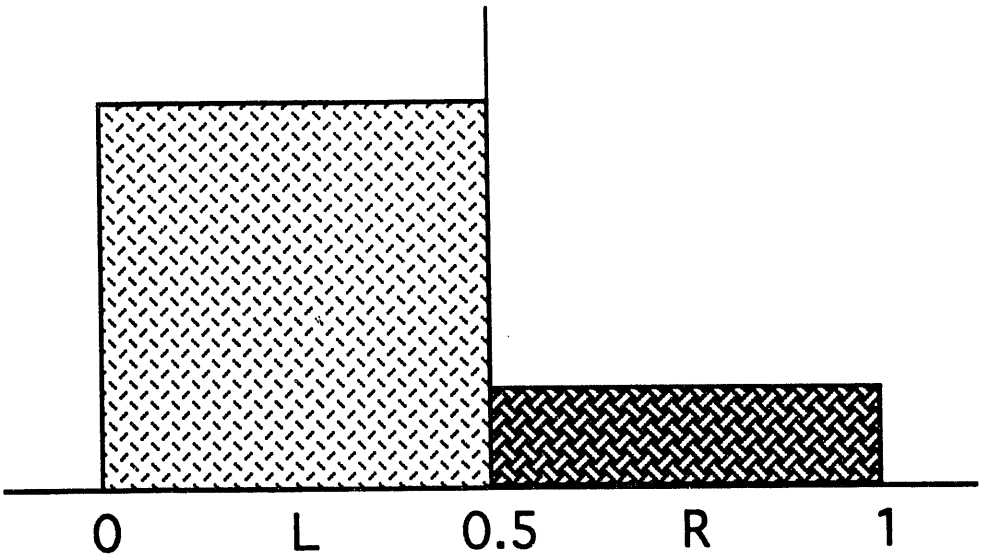


Figure 6-1. Initial conditions for hydrodynamic shock tube problem.

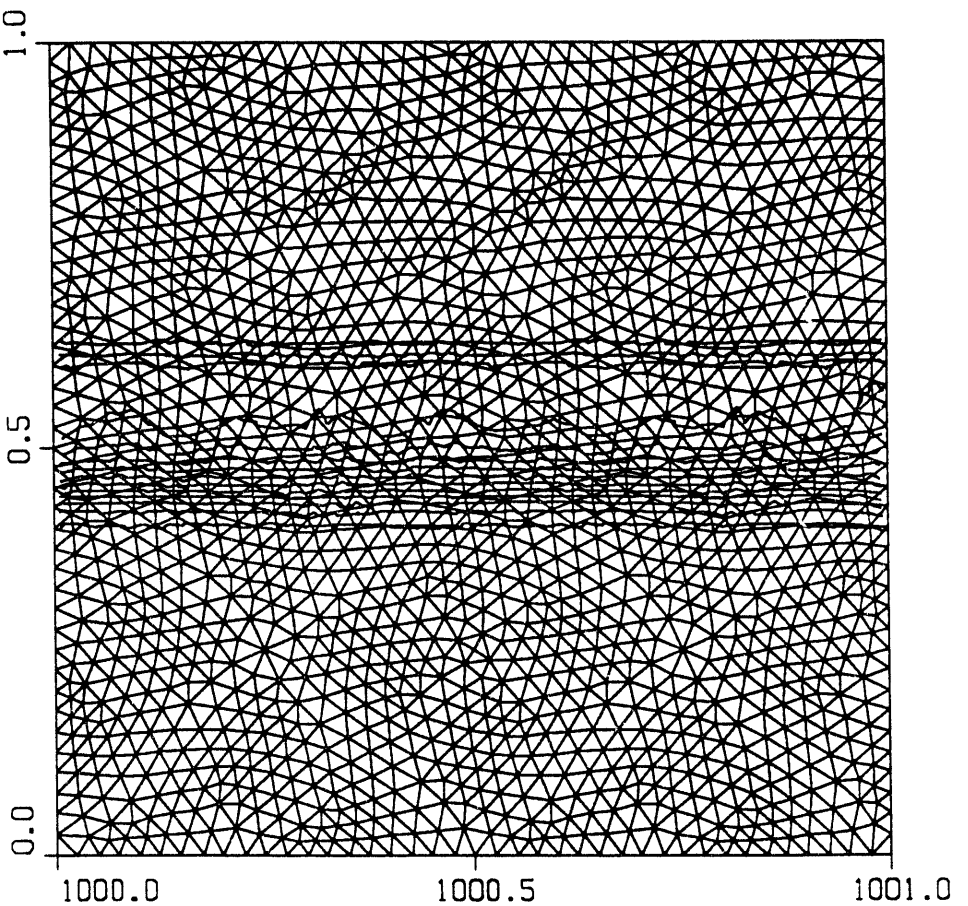
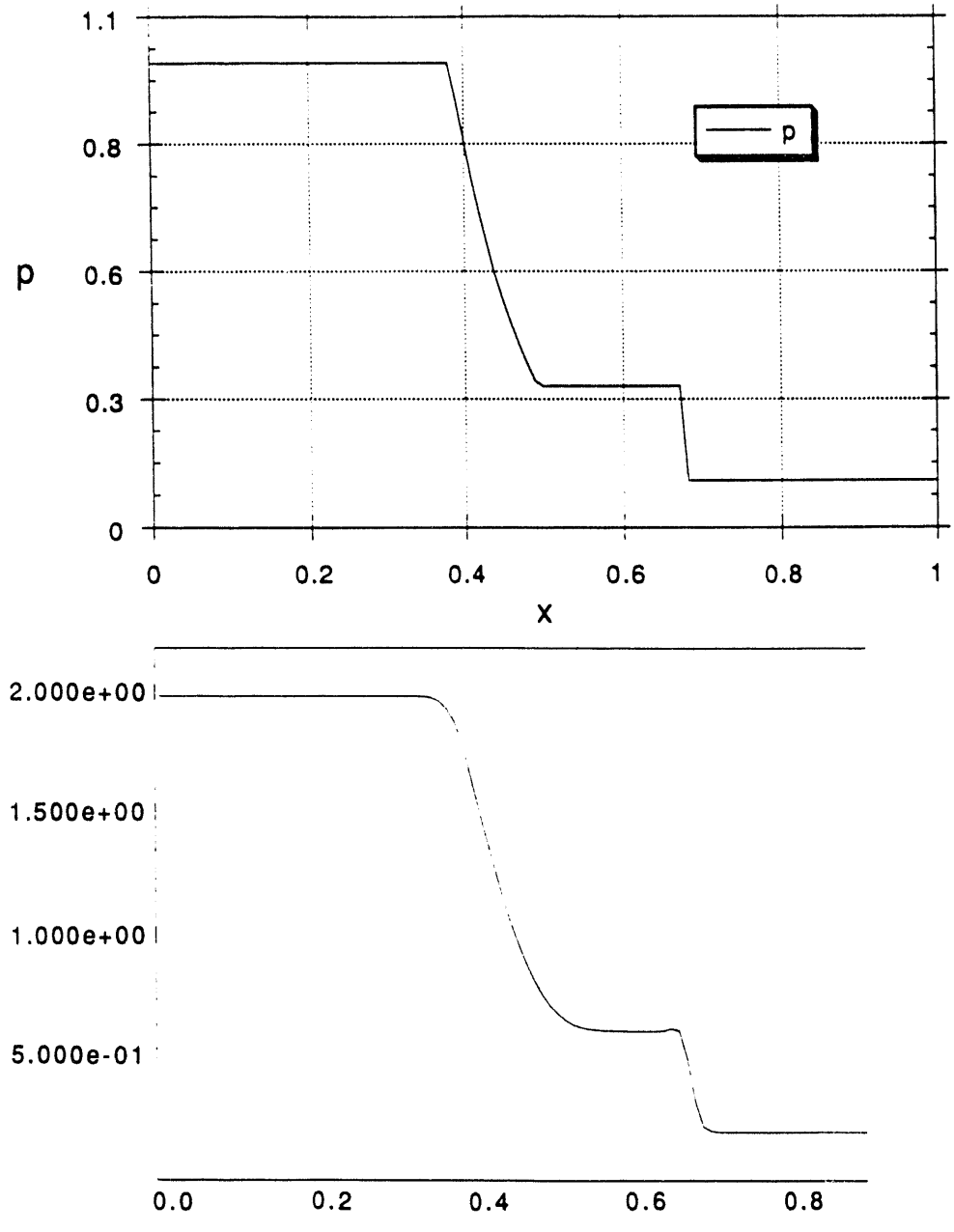
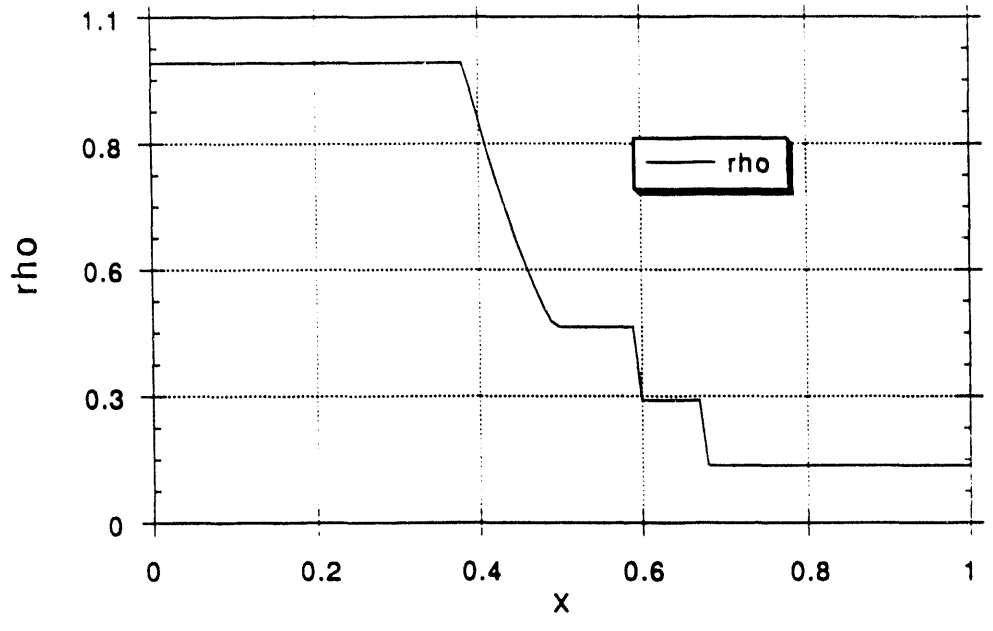


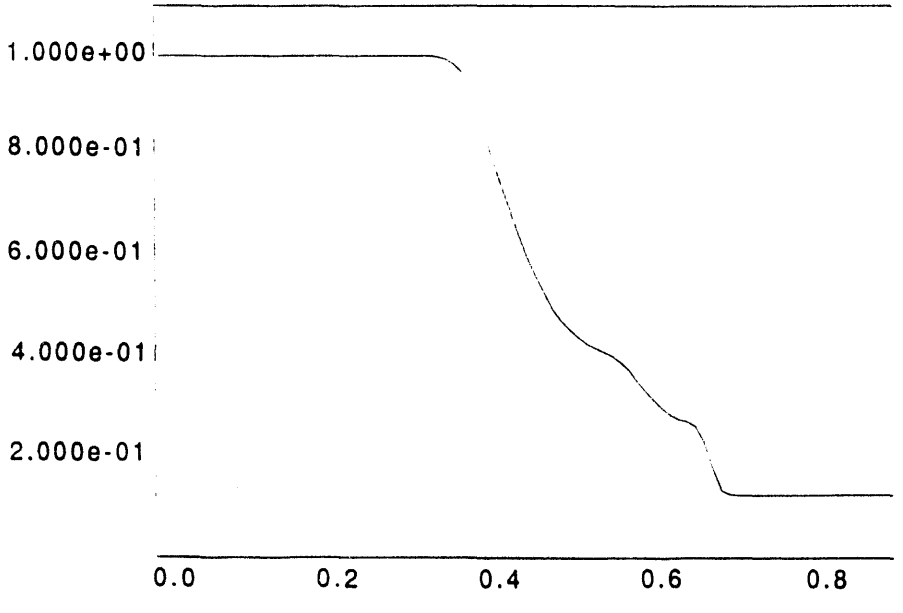
Figure 6-2. Mesh for hydrodynamic shock tube problem, with superimposed countours.



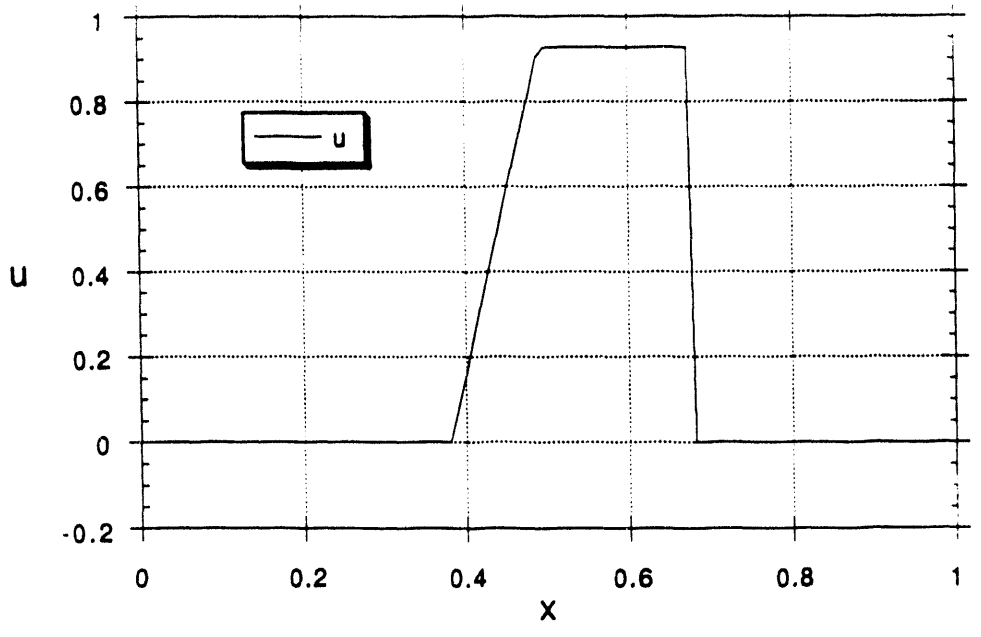
(a)

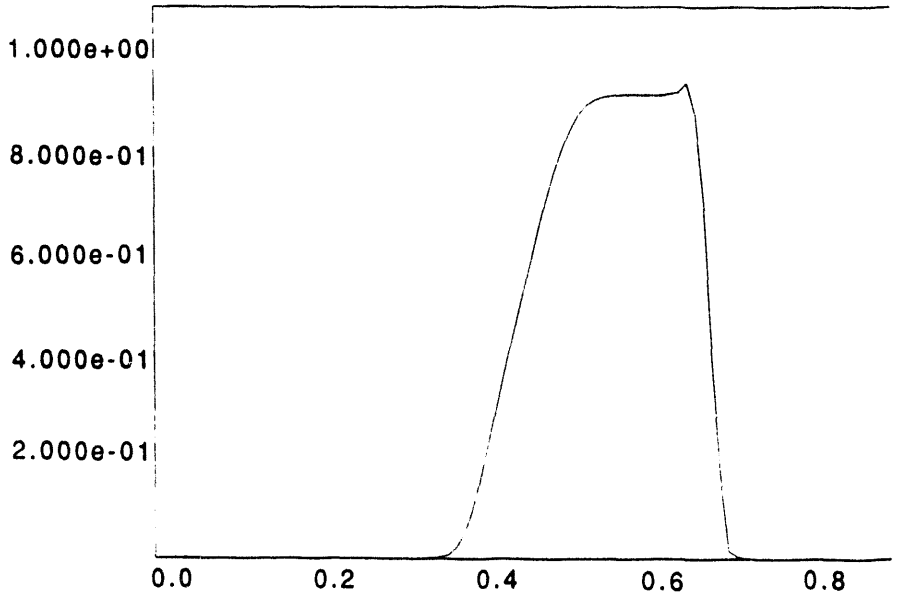
Figure 6-3a-c. Comparison of numerical and analytic solutions of the hydrodynamic shock tube problem at $t = 0.1$.


1.1
0.8
0.6
0.3
0
rho
rho
0
0.2
0.4
0.6
0.8
1
X


1.000e+00
8.000e-01
6.000e-01
4.000e-01
2.000e-01
0.0
0.2
0.4
0.6
0.8

(b)


u
1
0.8
0.6
0.4
0.2
0
-0.2
u
0
0.2
0.4
0.6
0.8
1
X


1.000e+00
8.000e-01
6.000e-01
4.000e-01
2.000e-01
0.0
0.2
0.4
0.6
0.8

(c)

One solution to the problem of numerical diffusion is to employ a higher order approximation to the interface fluxes. Another solution is to use a low order method but to reduce the diffusion by adaptively refining the mesh in the regions near the discontinuities. We have applied the mesh refinement techniques described in Section 2.7 to this problem. The adaptively refined mesh is shown in Figure 6-4. In Figure 6-5a-c we compare the solution at $t = 0.1$ with and without adaption. Mesh adaption reduces the diffusion near the right-propagating shock wave. The effect on the contact discontinuity is less satisfactory. A more accurate treatment of this structure likely requires higher order methods.

We emphasize that problems involving strong shocks are uncommon in fusion plasmas, so that low-order methods are sufficient for these applications. Nonetheless, it is desirable to develop an algorithm that is more universally applicable to a variety of problems. The control of unwanted numerical diffusion will be the goal of some of our future efforts.

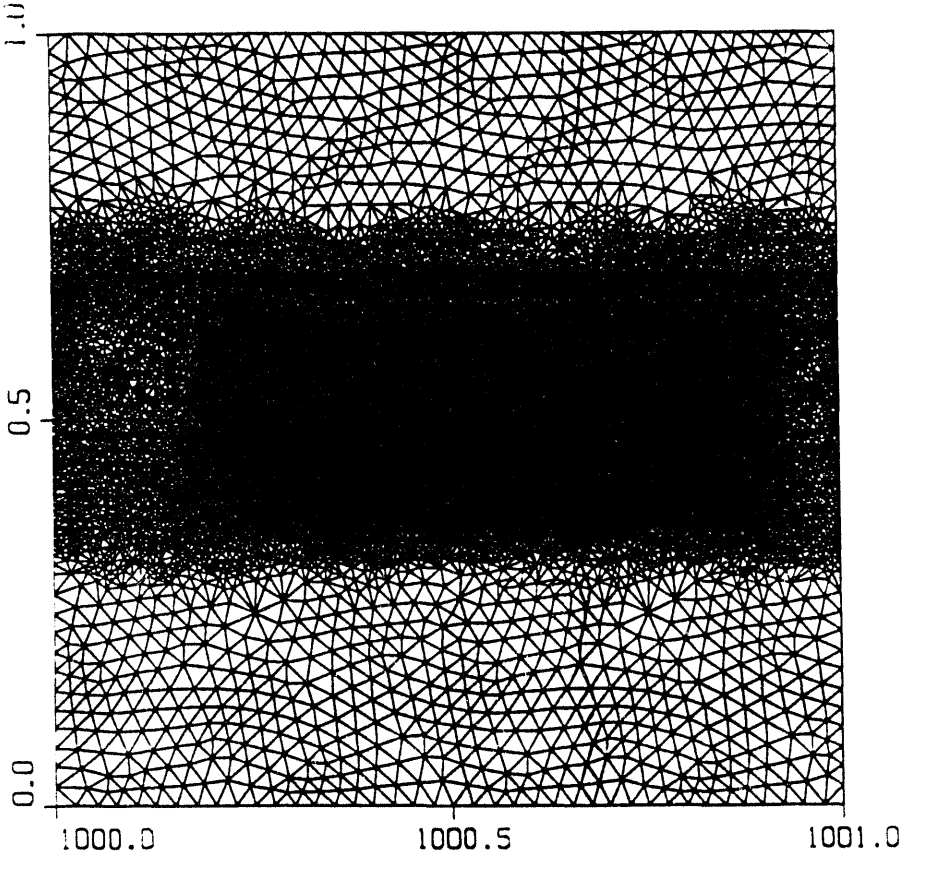
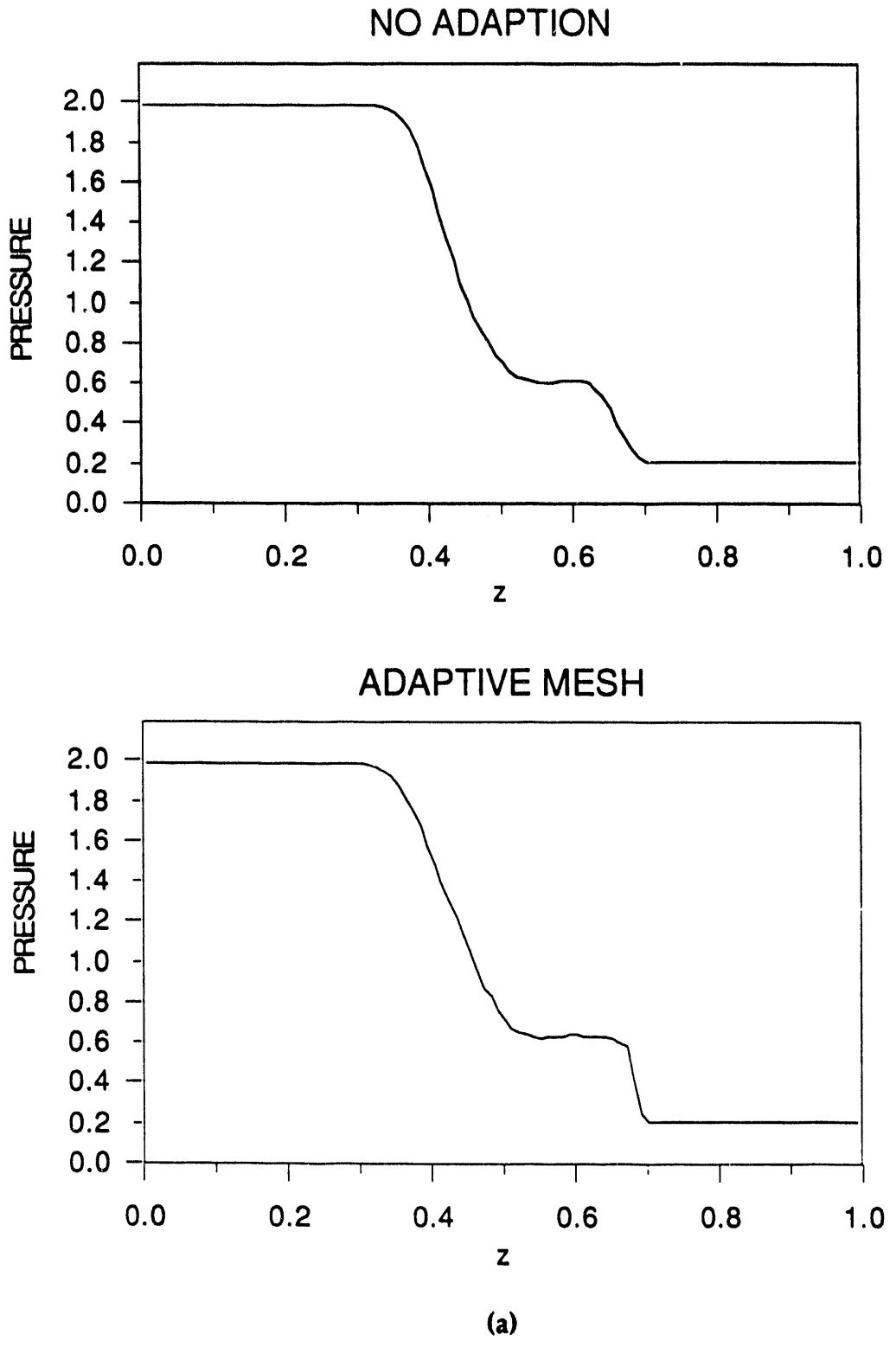
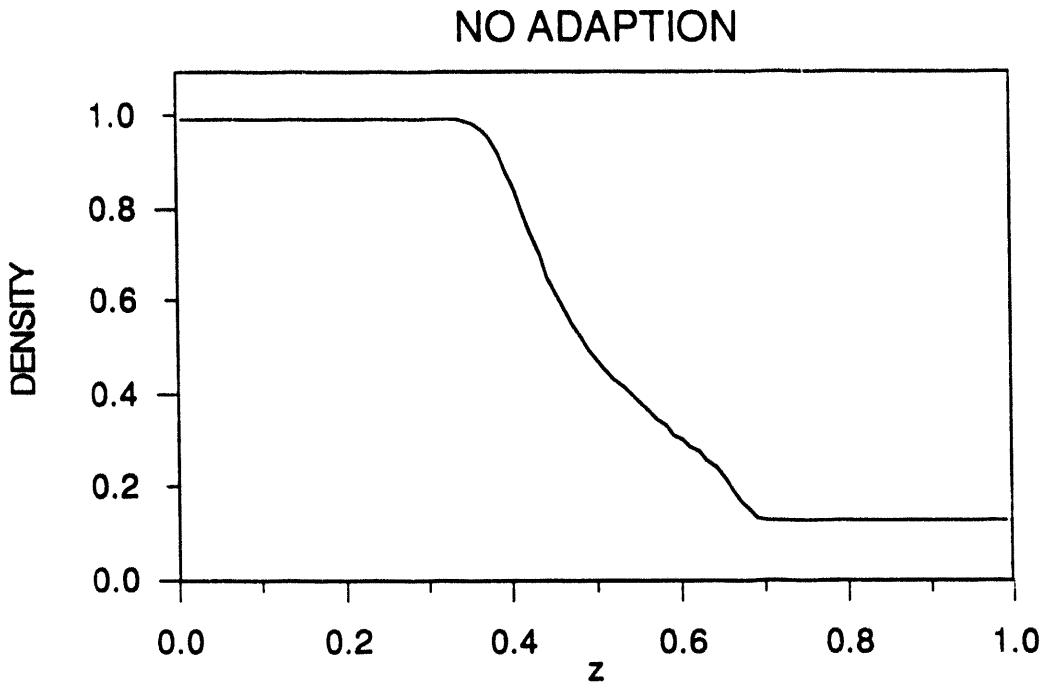


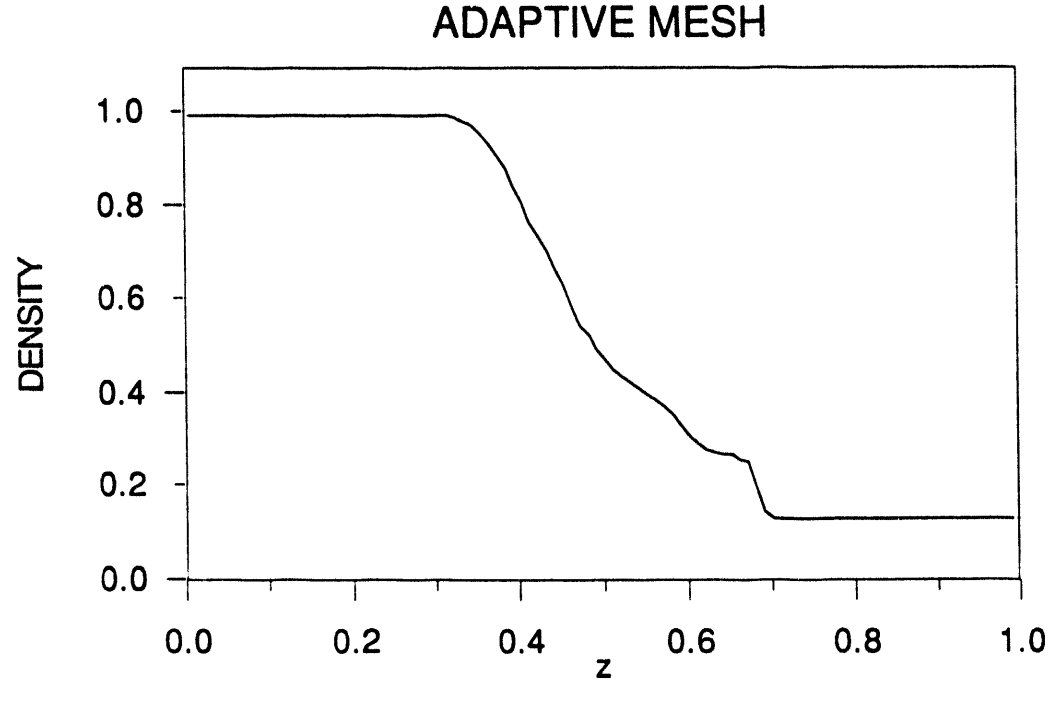
Figure 6-4. Adaptively refined mesh for the hydrodynamic shock tube problem.



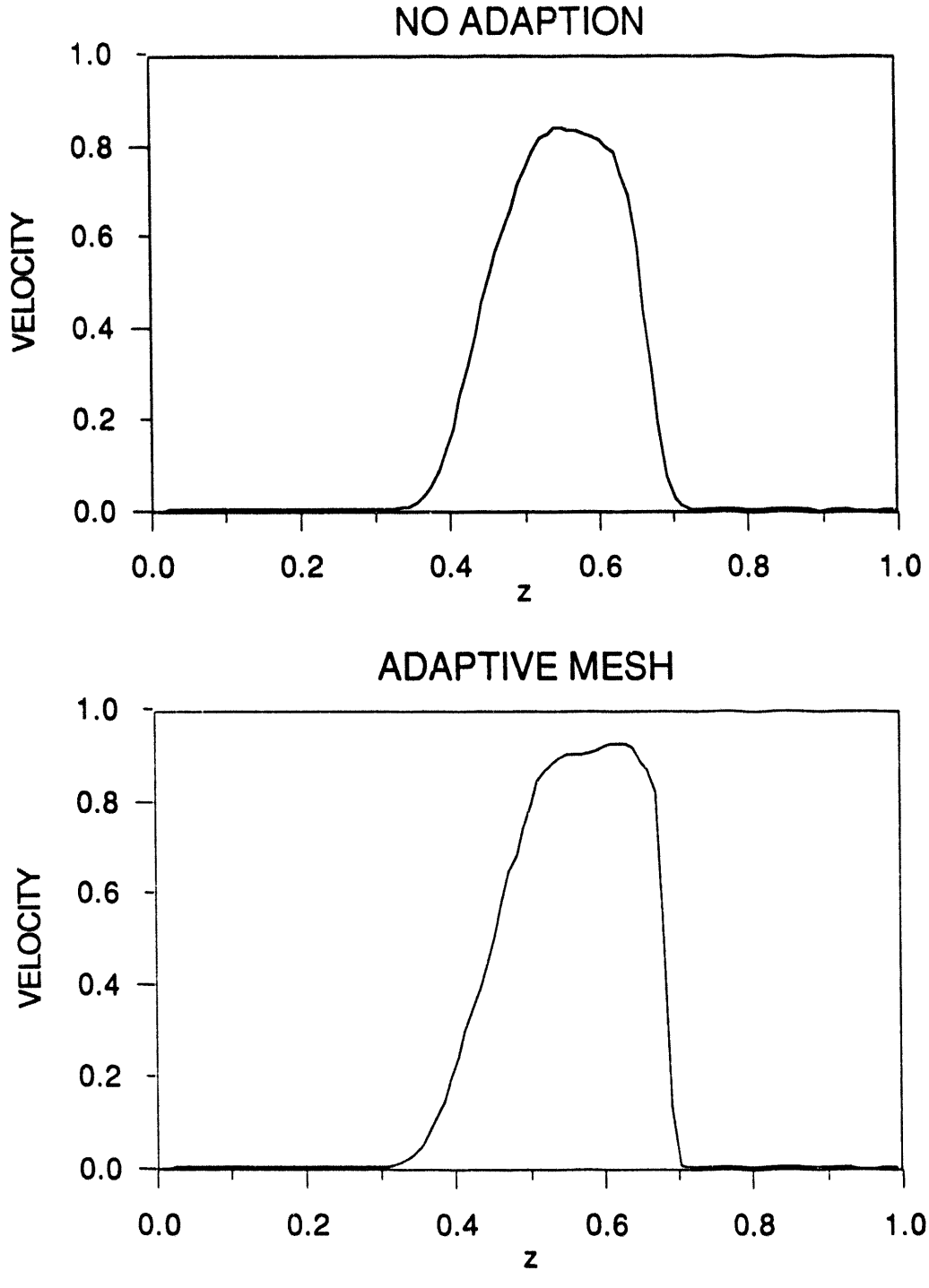
(a)

Figure 6-5a-c. Comparison of numerical solutions of the hydrodynamic shock tube problem with and without adaptive mesh refinement.


NO ADAPTION
DENSITY
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
z


ADAPTIVE MESH
DENSITY
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
z

(b)


NO ADAPTION
VELOCITY
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
z
ADAPTIVE MESH
VELOCITY
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
z

(c)

6.2 The Magnetohydrodynamic Shock Tube Problem

The hydrodynamic shock tube solution described in the pervious section has been extended to MHD by Brio and Wu[16]. The thermodynamic properties of the left and right states are the same is in the purely hydrodynamic case. A uniform magnetic field B_x is imposed in the direction (x) perpendicular to the membrane. The component of the magnetic field B_y parallel to the membrane is discontinuous at the membrane, with $B_{yL} = 1$ and $B_{yR} = -1$. The membrane is thus a current sheet in the z-direction. The magnetic configuration is sketched in Figure 6-6.

The dynamics after the membrane is ruptured are much more complex than in the purely hydrodynamic case; we refer the reader to Reference 15 for details. In Figures 6-7a-e we compare our two-dimensional solution with the more finely resolved one-dimensional solution of Brio and Wu[15]. We find that most of the details of the Brio-Wu solution are reproduced in our results, although the effect of the low-order diffusion is again apparent, especially near the contact discontinuity. We have also repeated the calculation with the component of magnetic field parallel to the membrane rotated by $\pi/2$, and find identical results for this polarization.

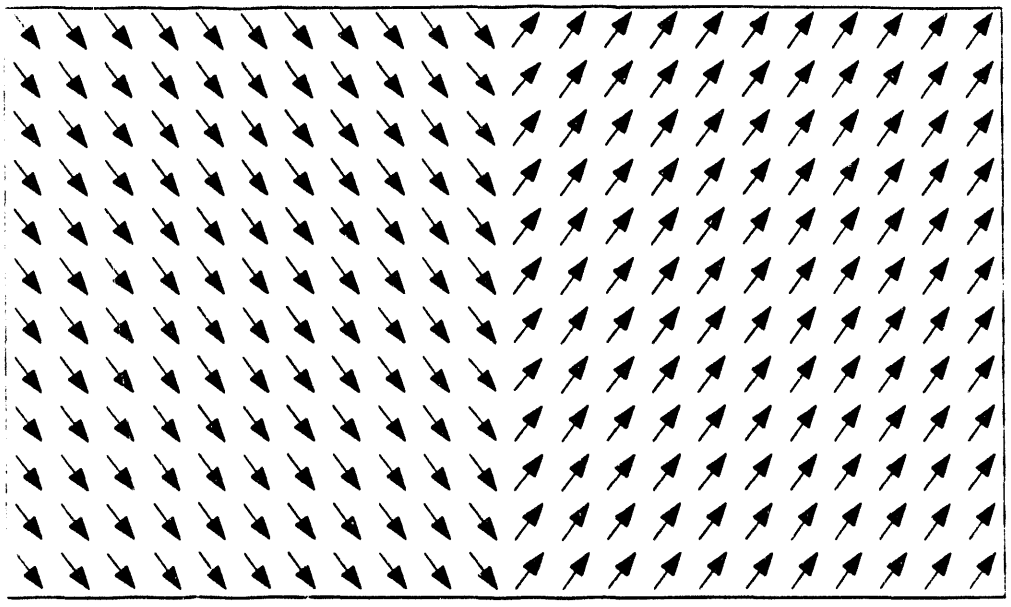
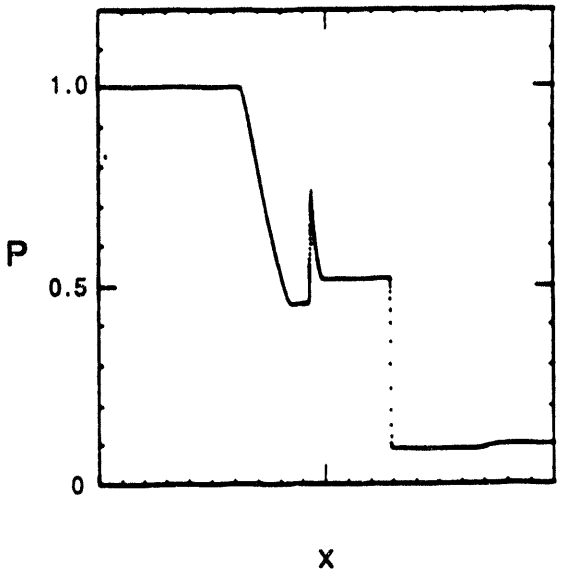
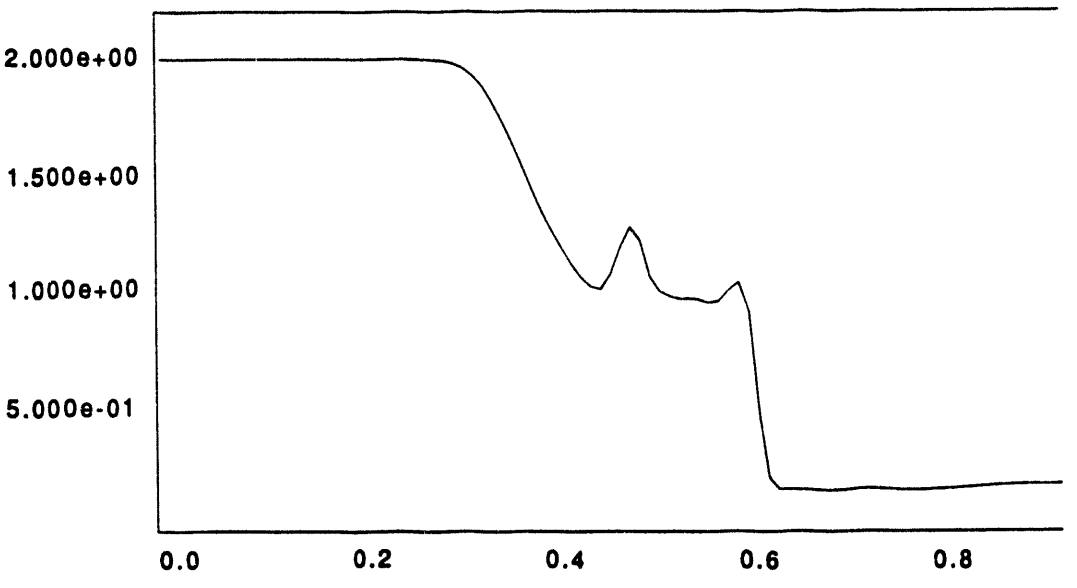


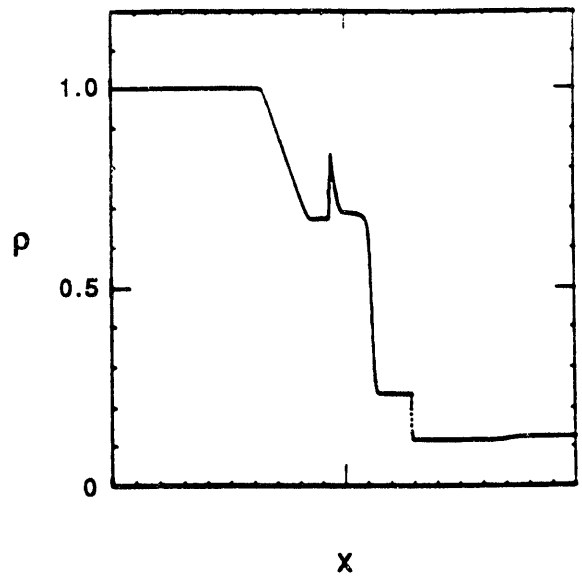
Figure 6-6. Initial magnetic field vectors for the MHD shock tube problem.

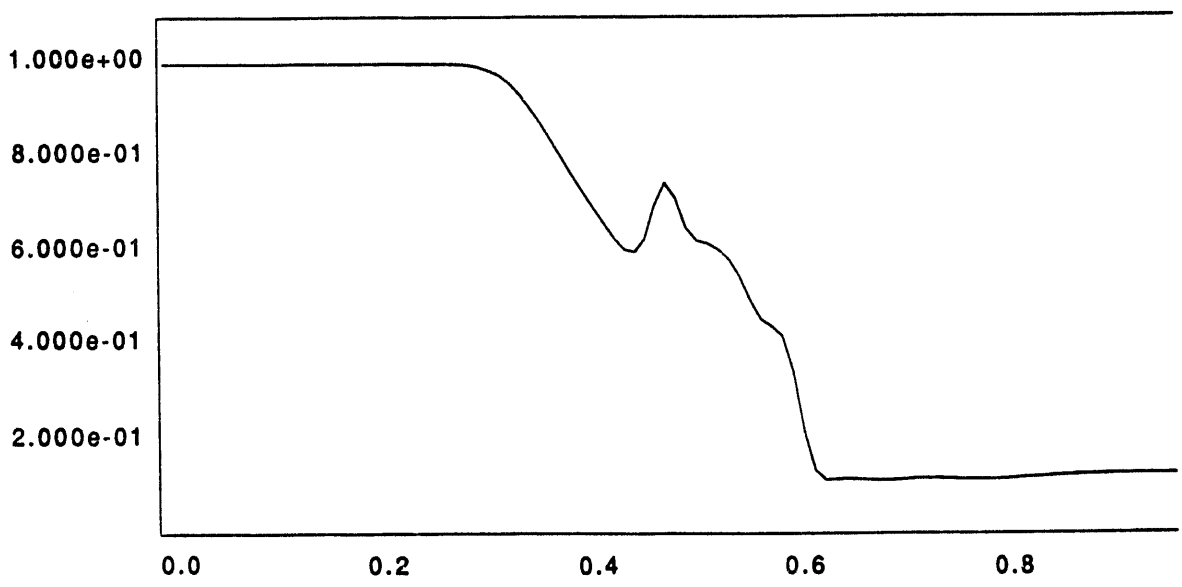




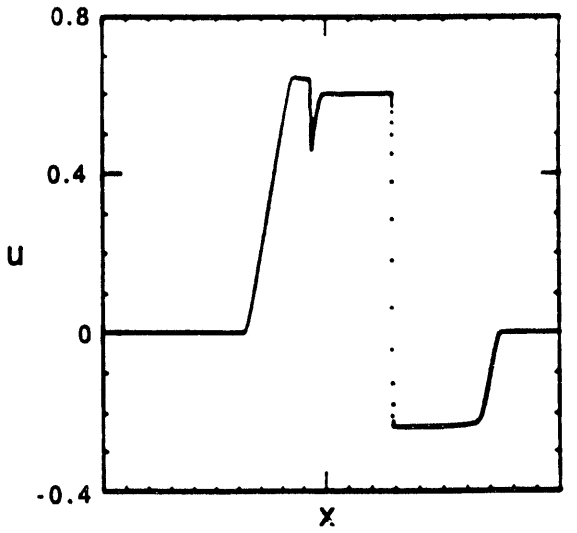
(a)

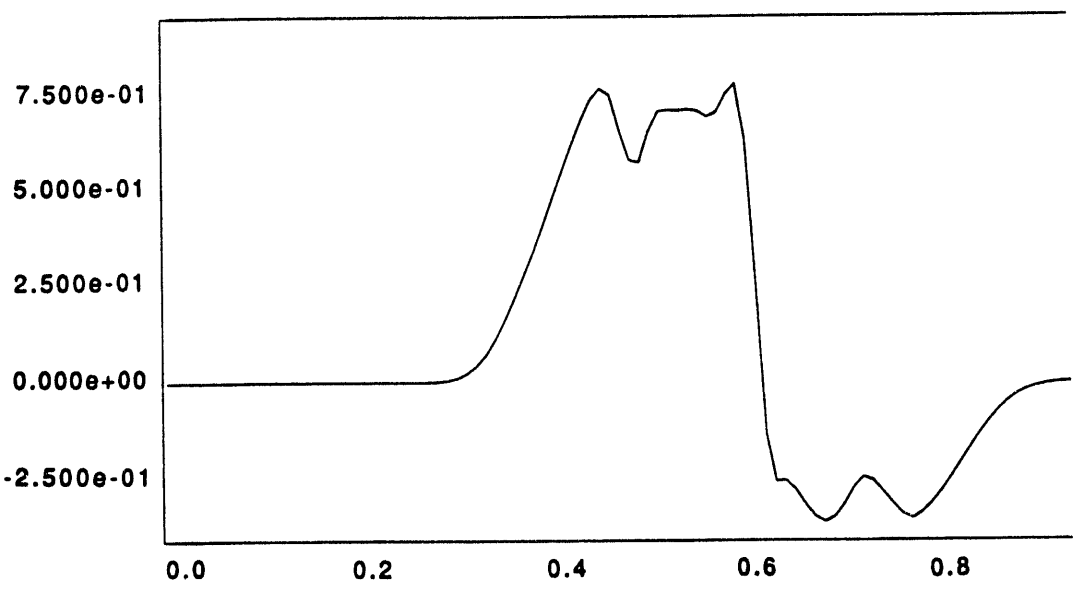
Figure 6-7a-e. Comparison of numerical solution of the MHD shock tube problem with the solution given in Reference 15.


1.0
0.5
0
ρ
x

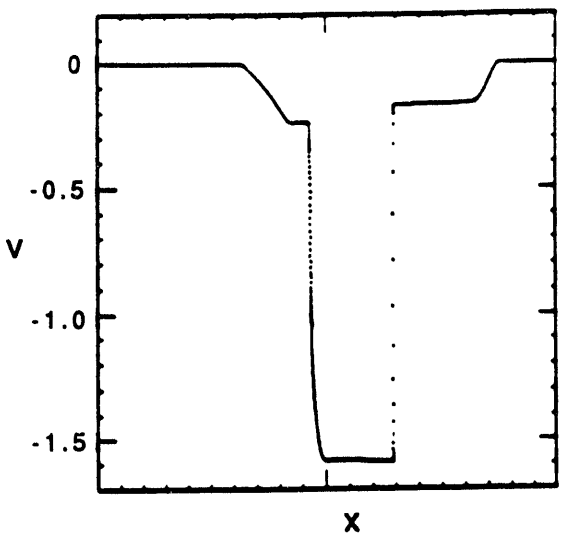

1.000e+00
8.000e-01
6.000e-01
4.000e-01
2.000e-01
0.0
0.2
0.4
0.6
0.8

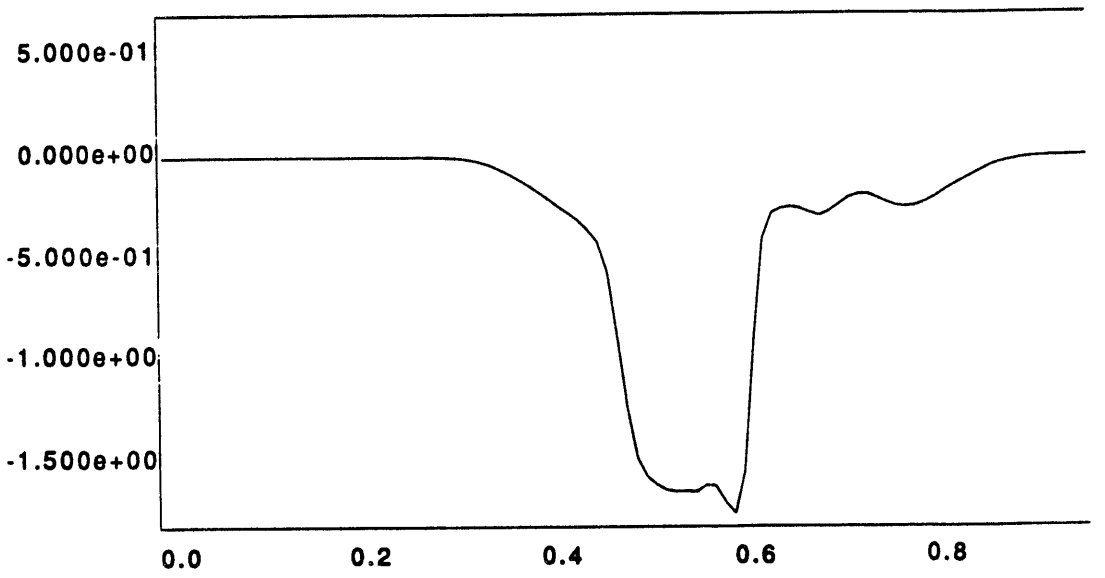
(b)


0.8
0.4
u
0
-0.4
x

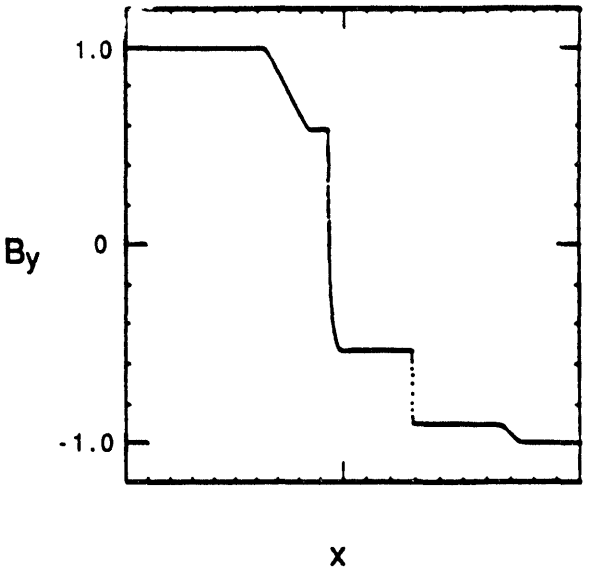

7.500e-01
5.000e-01
2.500e-01
0.000e+00
-2.500e-01
0.0
0.2
0.4
0.6
0.8

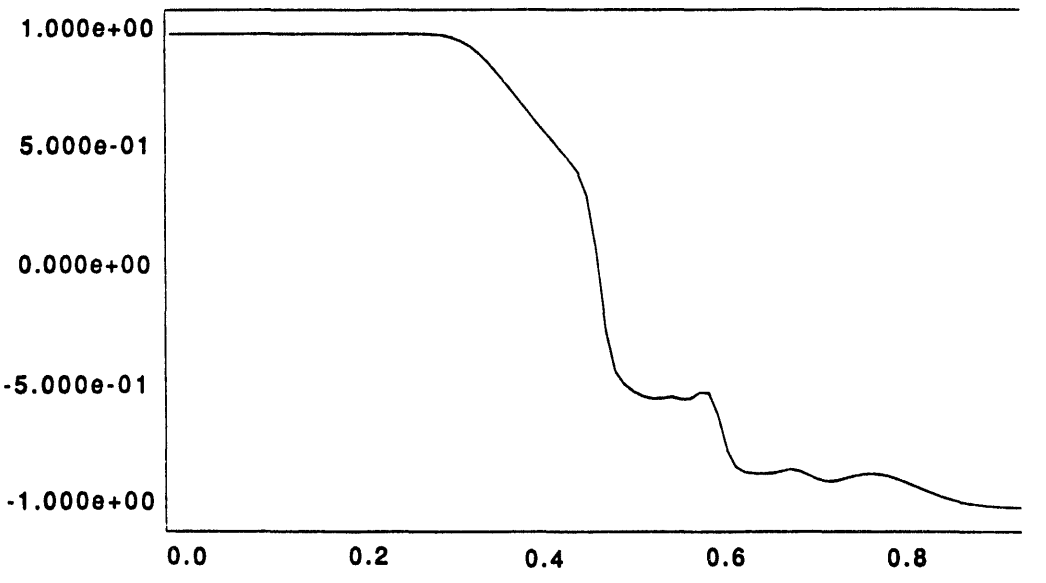
(c)


0
-0.5
V
-1.0
-1.5
X


5.000e-01
0.000e+00
-5.000e-01
-1.000e+00
-1.500e+00
0.0
0.2
0.4
0.6
0.8

(d)


1.0
By
0
-1.0
x


1.000e+00
5.000e-01
0.000e+00
-5.000e-01
-1.000e+00
0.0
0.2
0.4
0.6
0.8

(e)

We have not yet applied adaptive mesh techniques to this MHD problem because of errors encountered in interpolating electromagnetic quantities to the new mesh. This is symptomatic of the problems discussed at the end of Section 4.3. We are actively investigating solutions to this problem.

6.3 Toroidal Equilibria

To be useful for magnetic fusion applications the TRIM algorithm must be able to describe force balance in low aspect ratio toroidal systems. Force balance is given by solutions to the Grad-Shafranov equation

$$\Delta^* \psi = r^2 \nabla \cdot \frac{\nabla \psi}{r^2} = \frac{1}{2} r^2 \frac{dP}{d\psi} - F \frac{dF}{d\psi} \tag{6.1}$$

where $\psi(r,z) = rA_\phi$ is the poloidal flux, and the pressure $P(\psi)$ and the toroidal flux function $F(\psi) = rB_\phi$ are arbitrary functions of ψ. An analytic solution has been given by Solov'ev[16]. With

$$P(\psi) = -\frac{4}{\varepsilon^2} \frac{1+\kappa^2}{\kappa^2} (\psi - 1) \tag{6.2}$$

$$F(\psi) = \frac{2b}{\varepsilon\kappa} (1 - \psi)^{1/2} + C \tag{6.3}$$

the poloidal flux and toroidal field are

$$\psi(r,z) = \frac{1}{\varepsilon^2} \left[\frac{\left(r^2 + b^2\right) z^2}{\kappa^2} + \frac{\left(r^2 - 1\right)^2}{4} \right] \tag{6.4}$$

$$B_\phi = \frac{C}{r} + O(b) \tag{6.5}$$

where $\varepsilon = a/R$ is the inverse aspect ratio, κ is the elongation, b is a diamagnetic factor, and C is a normalization constant that determines the strength of the vacuum toroidal field. Contours of ψ and P with $\kappa = 1$ and $b = 0$ are shown in Figure 6-8. Since $b = 0$, this equilibrium has no poloidal current ($J_r = J_z = 0$).

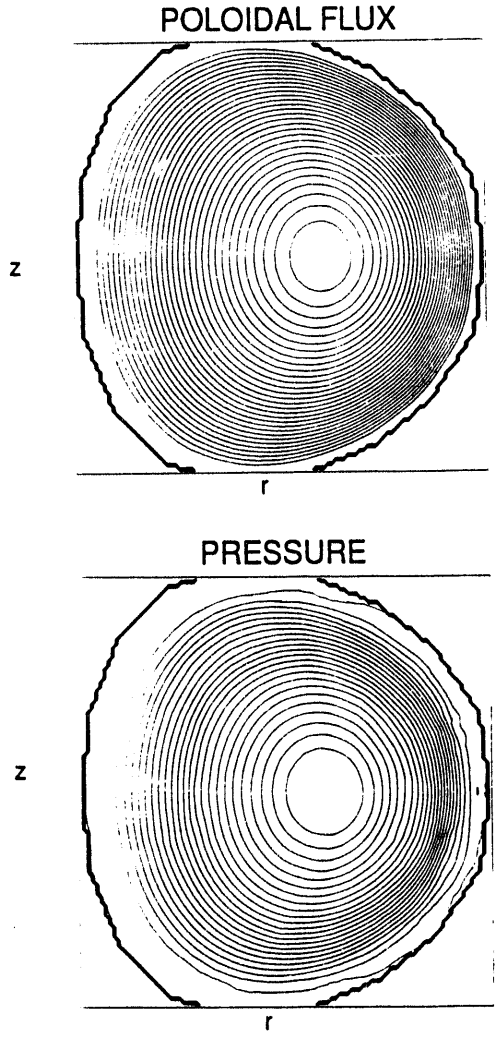


Figure 6-8. Contours of poloidal flux and pressure for the Solov'ev toroidal equilibrium.

A typical unstructured mesh for this problem with N_s = 1658 is shown in Figure 6-9. The outer boundary corresponds to $\psi = 1$. When Equations (6.2), (6.4), and (6.5) are introduced onto this mesh, the right hand side of Equation (4.5) (the momentum equation) is a small number of the order of the truncation error of the finite volume approximation: there are unbalanced forces to this order. This force imbalance excites Alfvén and sound waves. In Figure 6-10 we plot the kinetic energy of these oscillations versus time for two values of the viscosity. The viscosity effectively removes these modes and the system finds a neighboring state of forces balance on the unstructured mesh.

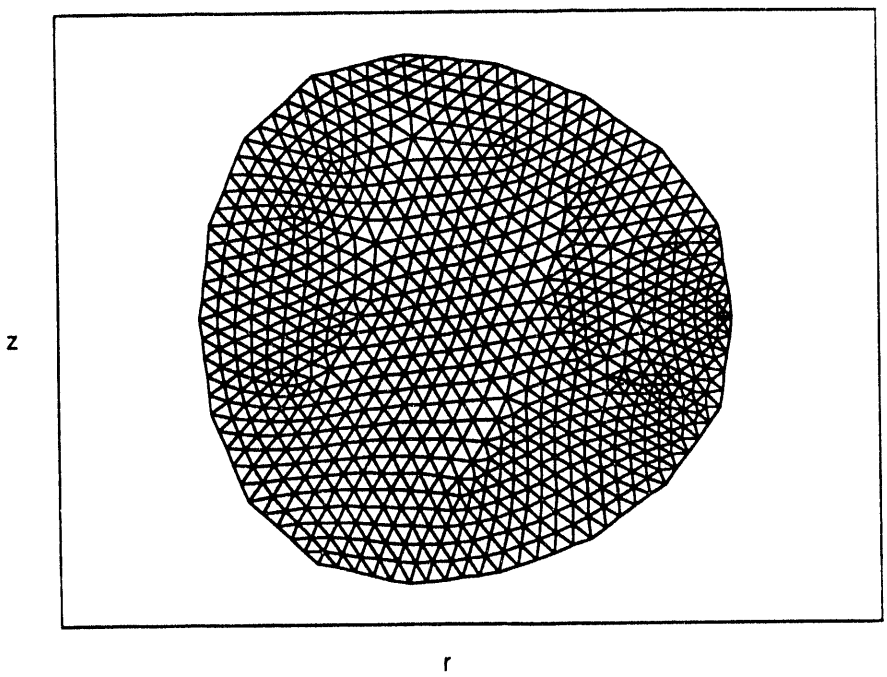


Figure 6-9. Unstructured mesh for the Solov'ev equilibrium.

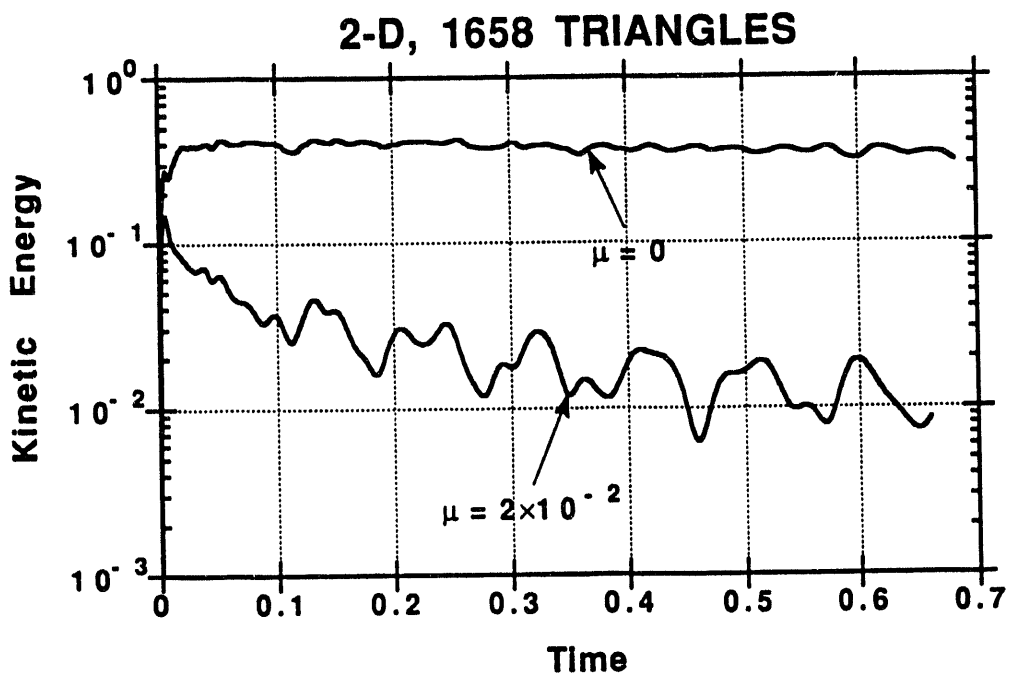


Figure 6-10. Kinetic energy versus time during relaxation of the Solov'ev equilibrium.

The averaging procedures described in Section 4.3 introduce diffusion in the magnetic energy due to the velocity averages used in Ohm's law. In Figure 6-11 we plot the magnetic energy as a function of time for cases with and without viscosity. The lower velocity in the viscous case causes the magnetic damping to decrease. This damping is also affected by the number of triangles in the mesh, N_s, as illustrated in Figure 6-12 for cases with three different values of N_s. This damping rate is summarized in Figure 6-13. We see that the numerical damping rate is approximately linear in $\Delta \approx (\Delta a_s)^{1/2}$, confirming the first order nature of the algorithm.

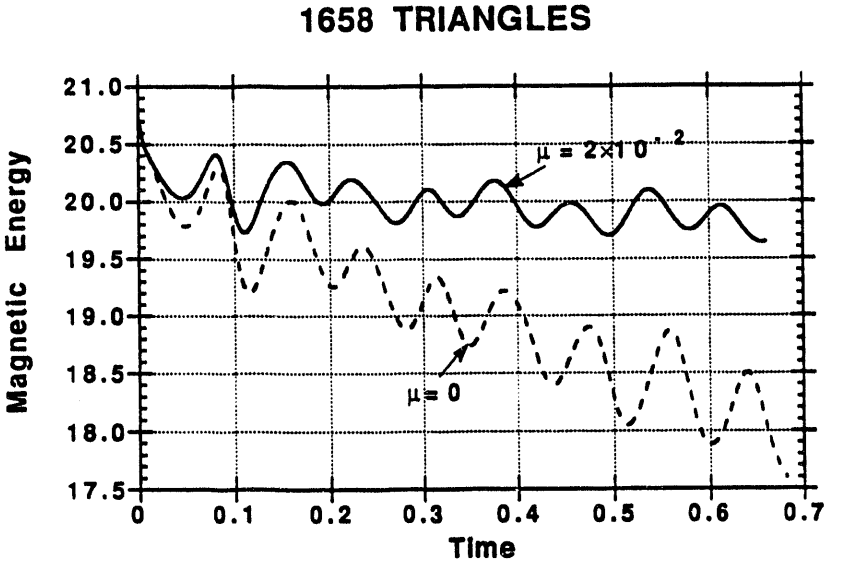


Figure 6-11. Magnetic energy versus time during relaxation of the Solov'ev equilibrium.

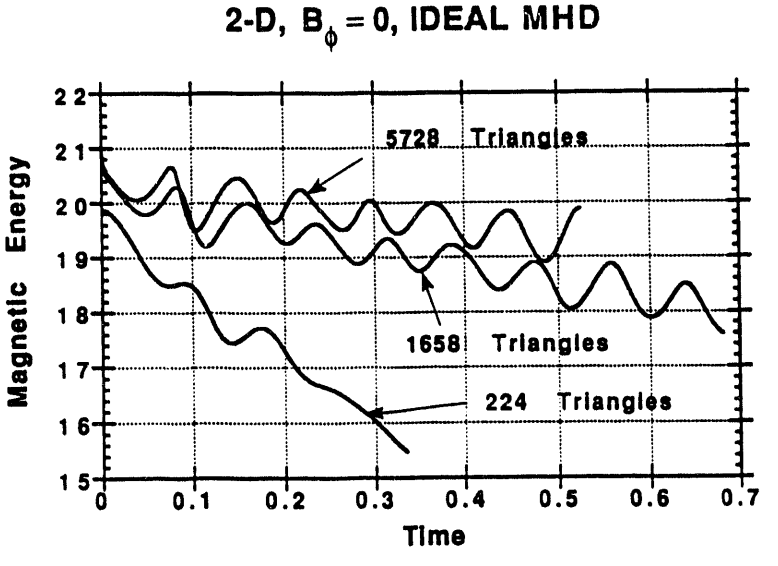


Figure 6-12. Magnetic energy versus time during relaxation of the Solov'ev equilibrium for three different meshes.

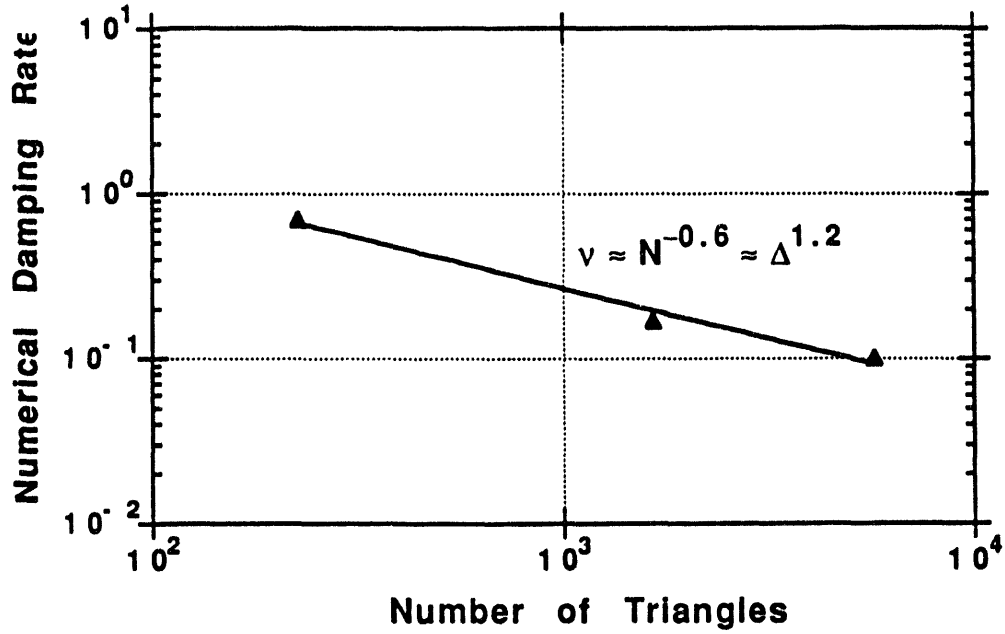


Figure 6-13. Numerical damping rate versus number of triangles.

6.4 Nonlinear Evolution of Toroidal Instabilities

One of the standard applications of MHD to fusion plasmas is the linear growth and nonlinear saturation of instabilities. These instabilities can occur because toroidal equilibria of the type described in Section 6.3 are not necessarily minimum energy states, even though they are extrema of the energy. Equilibria that are local maxima of the energy are unstable, with deviations from the initial state growing exponentially in time. Determining the stability of equilibria is an important problem in the design of a fusion experiment, and even stable equilibria can be driven unstable by diffusive processes[5]. The growth rate of an instability can be found by solving a linear eigenvalue problem. The details of the saturation requires that a nonlinear, time-dependent problem be solved.

In Figure 6-14a,b we display poloidal flux and pressure contours for an equilibrium that is representative of ITER, an international fusion test reactor that is presently being designed. This equilibrium has both toroidal and poloidal currents, and has been obtained by a careful numerical solution[17] of Equation (70). The outer boundary is the separatrix, or last closed flux surface; it is intended that the plasma is confined within this surface. The functions $P(\psi)$ and $F(\psi)$ are displayed in Figure 6-15. This equilibrium has been constructed to be unstable to an internal kink mode with toroidal mode number $n = 1$. The safety factor profile is shown in Figure 6-16. The linear instability has been computed with the GATO code[18], which directly solves the resulting linear eigenvalue problem.

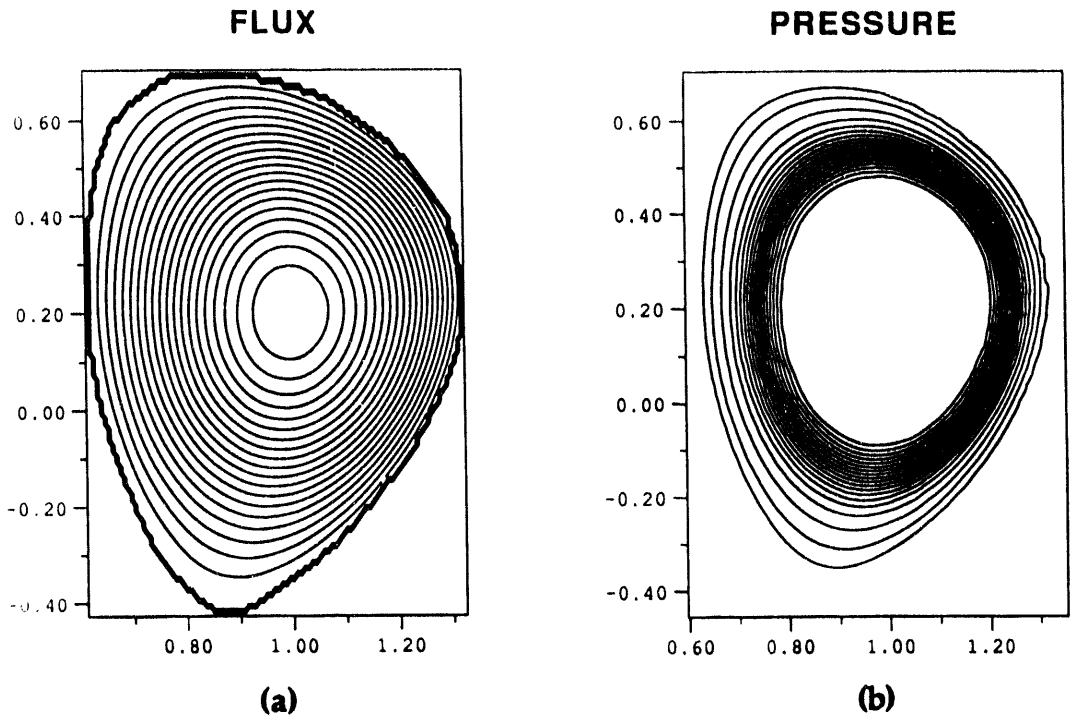


Figure 6-14a,b. Contours of poloidal flux and pressure for an ITER equilibrium.

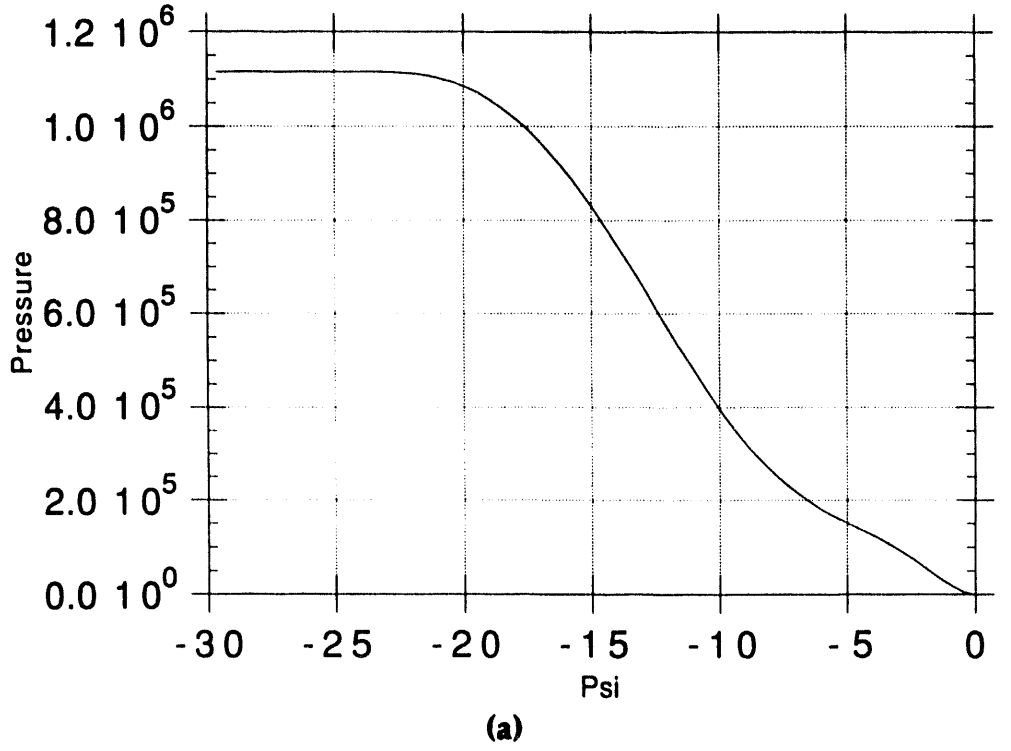
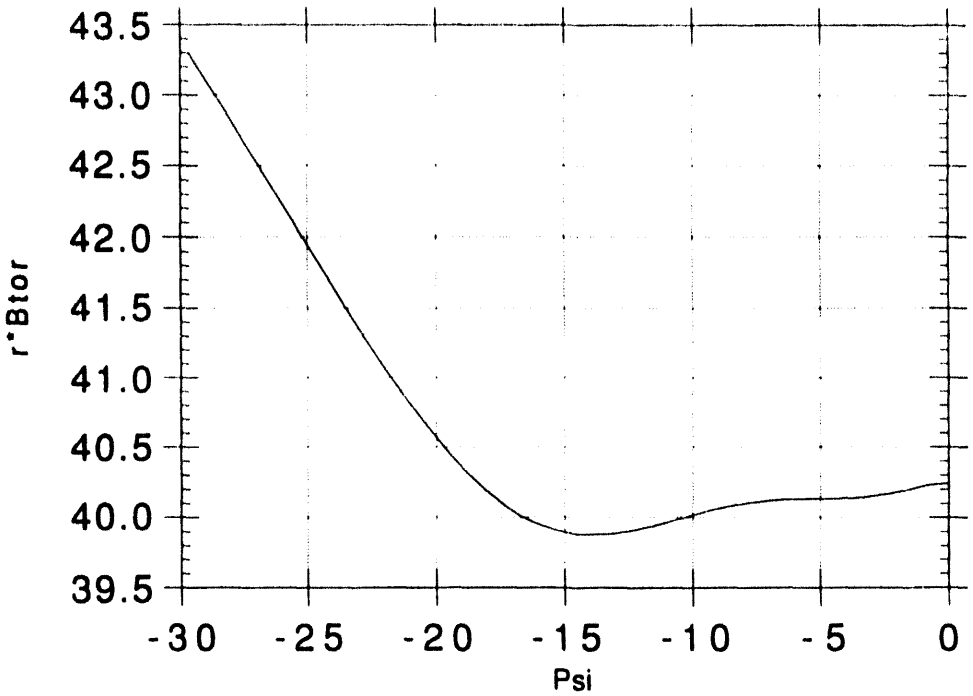


Figure 6-15a,b. The functions $F(\psi)$ and $P(\psi)$ for the ITER equilibrium.



(b)

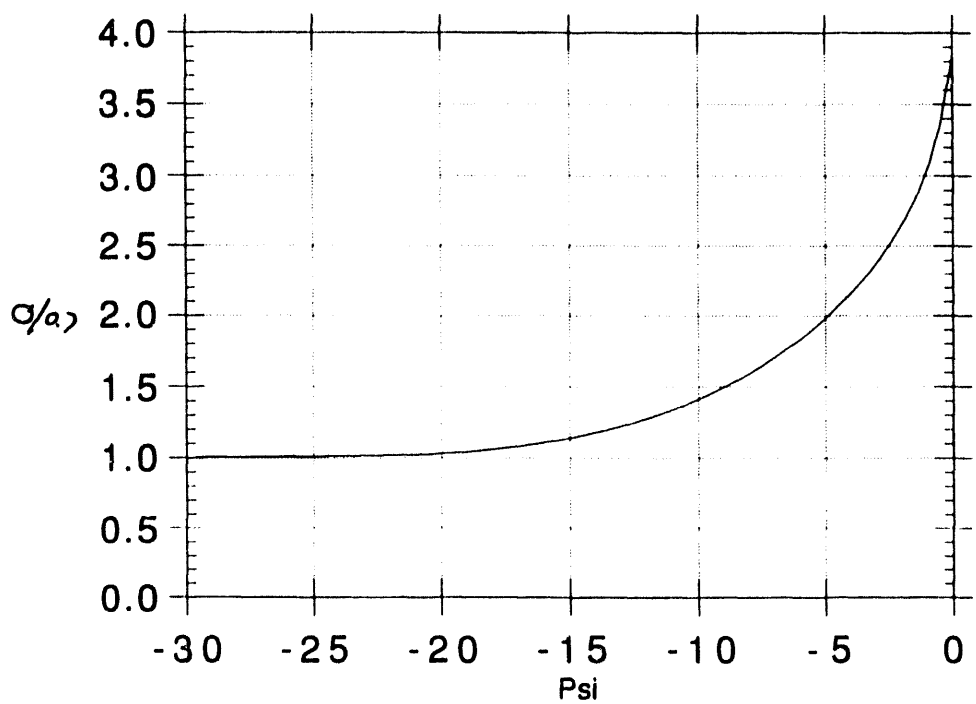


Figure 6-16. The safety factor q (ψ) for the ITER equilibrium.

The unstructured mesh inside the separatrix with N_s = 5728 is shown in Figure 6-17. The equilibrium is initialized to this mesh by cubic spline interpolation, and the resulting force imbalance is resolved with viscous damping as described in Section 6.3. We have also found it useful to introduce spatially dependent resistivity, with $S = 10^6$ near the magnetic axis and $S = 10^4$ near the separatrix. Thus resistive flow is always present and true static equilibrium is not achieved. The resistivity also causes the current to peak near the magnetic axis, thus altering the safety factor profile.

After axisymmetric relaxation, three-dimensional modes are perturbed with random noise at very low amplitude. For this calculation we use a toroidal mesh with $N_\phi = 8$ toroidal mesh points, so that three toroidal Fourier modes ($n = 0, 1, 2$) are included after dealiasing. This resolution is marginally acceptable for highly accurate calculations, but will demonstrate the utility of the TRIM algorithm for this problem. We expect that linearly unstable modes will begin to grow exponentially in time with linear growth rates comparable to those determined by GATO.

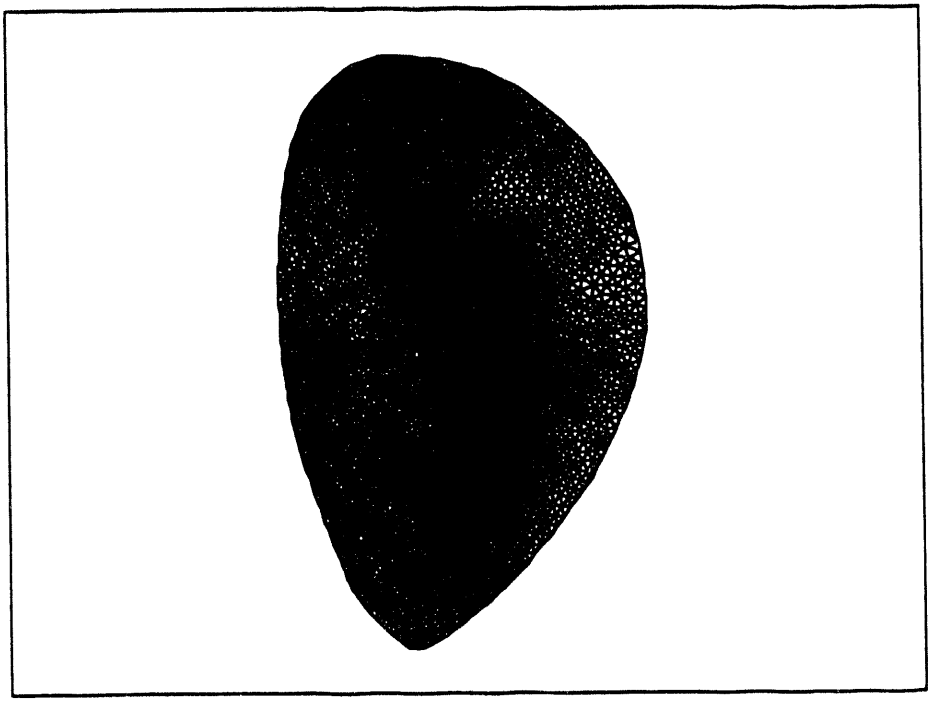


Figure 6-17. Unstructured mesh for the ITER equilibrium.

In Figure 6-18 we plot the kinetic and magnetic energy in all modes *except* $n = 0$ as a function of time. An exponentially growing solution is seen to arise from the initial random noise. This instability saturates nonlinearly at finite amplitude. The exponential growth rate is approximately a factor of 2 larger than that determined by GATO.

In Figure 6-19 we plot the magnetic and kinetic energy in the individual $n = 1$ and $n = 2$ modes as functions of time. It is seen that both $n = 1$ and $n = 2$ modes grow exponentially, with approximately the same linear growth rate. This is in contradiction to the results of GATO, which predicts stability for $n = 2$. However, both the appearance of the $n = 2$ mode and the enhanced growth rate of the $n = 1$ mode may be the result of the modified safety factor profile as a result of resistive diffusion. The $n = 2$ mode may also be nonlinear driven by the growing $n = 1$. These issues must be resolved in the near future.

The linear eigenmode for the poloidal velocity is shown in Figure 6-19, where poloidal velocity vectors at $t = 530$ are superimposed on contours of the poloidal flux at the eight toroidal locations included in the calculation. The flow pattern has the clear counter-rotating vortex structure of an internal kink mode with dominant poloidal mode number $m = 1$. This structure is seen to rotate once around the torus, as required by an $n = 1$ mode. As the mode saturates nonlinearly the structure takes on an $m = 2$, $n = 2$ character, as is seen in Figures 6-20 and 6-22. The three-dimensional structure of the pressure at saturation is shown in Figure 6-23.

The instabilities computed here evolve on a fraction of the poloidal Alfvén time, which is almost a factor of 10 longer than the toroidal Alfvén time. Purely explicit methods require that the time step be taken at a fraction of the shortest time scale. In the example computed here, we have used the semi-implicit method with a time step 30 times that allowed by explicit numerical stability. Clearly, this method is essential for computing tokamak instabilities with the primitive MHD equations.

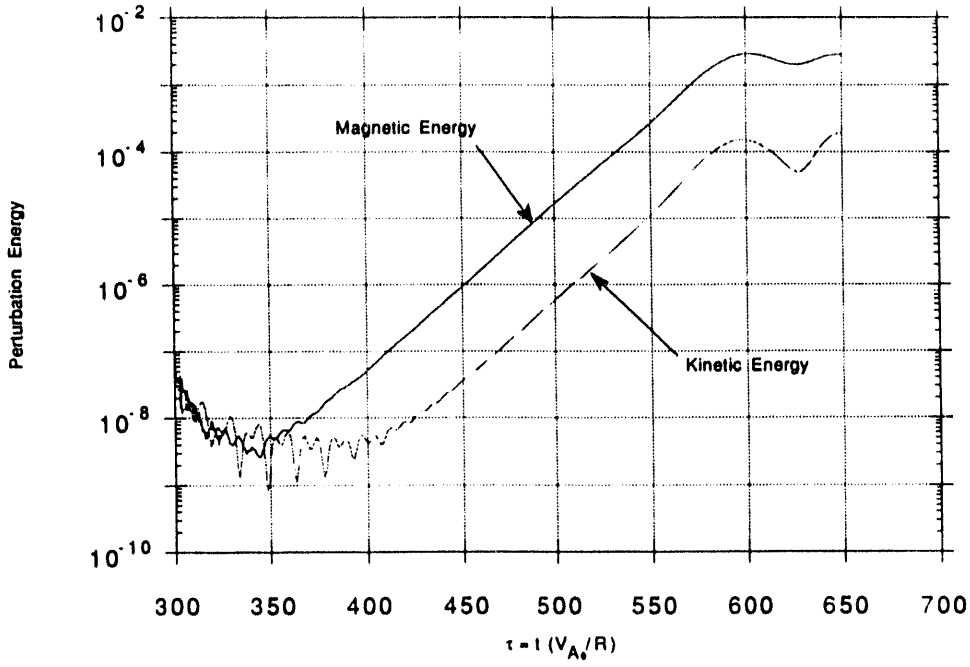


Figure 6-18. Kinetic and perturbed magnetic energies versus time for the evolution of an unstable $n = 1$ kink mode in the ITER equilibrium.

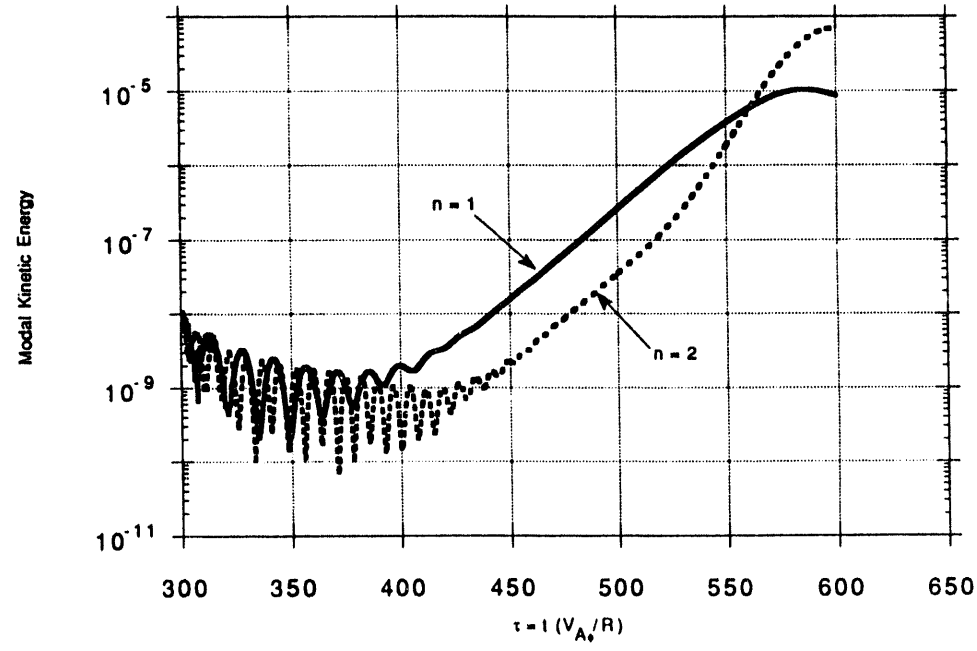


Figure 6-19. Magnetic and kinetic energies versus time for the $n = 1$ and $n = 2$ modes in the ITER equilibrium.

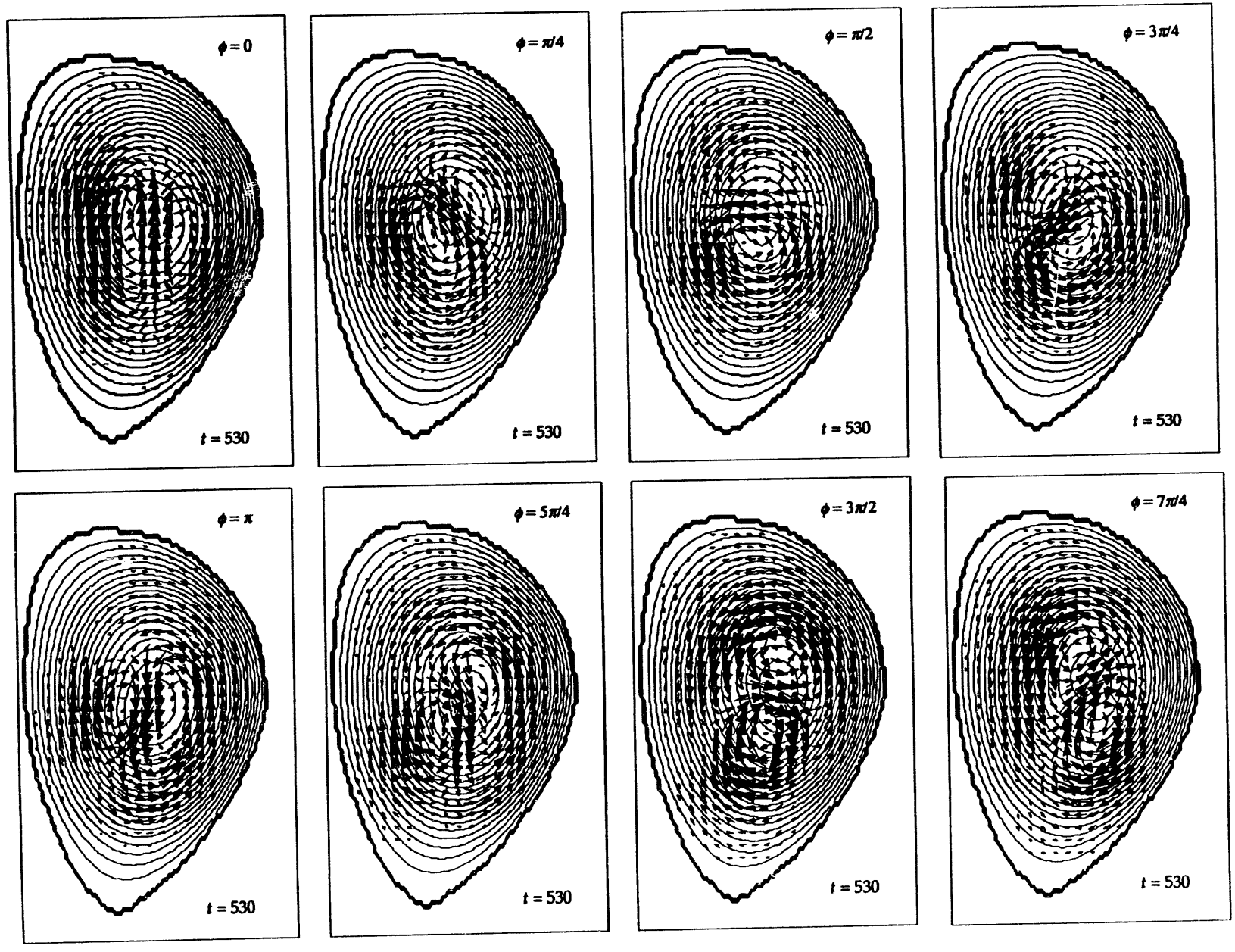


Figure 6-20. Velocity and poloidal flux at 8 different toroidal locations during the linear growth phase.

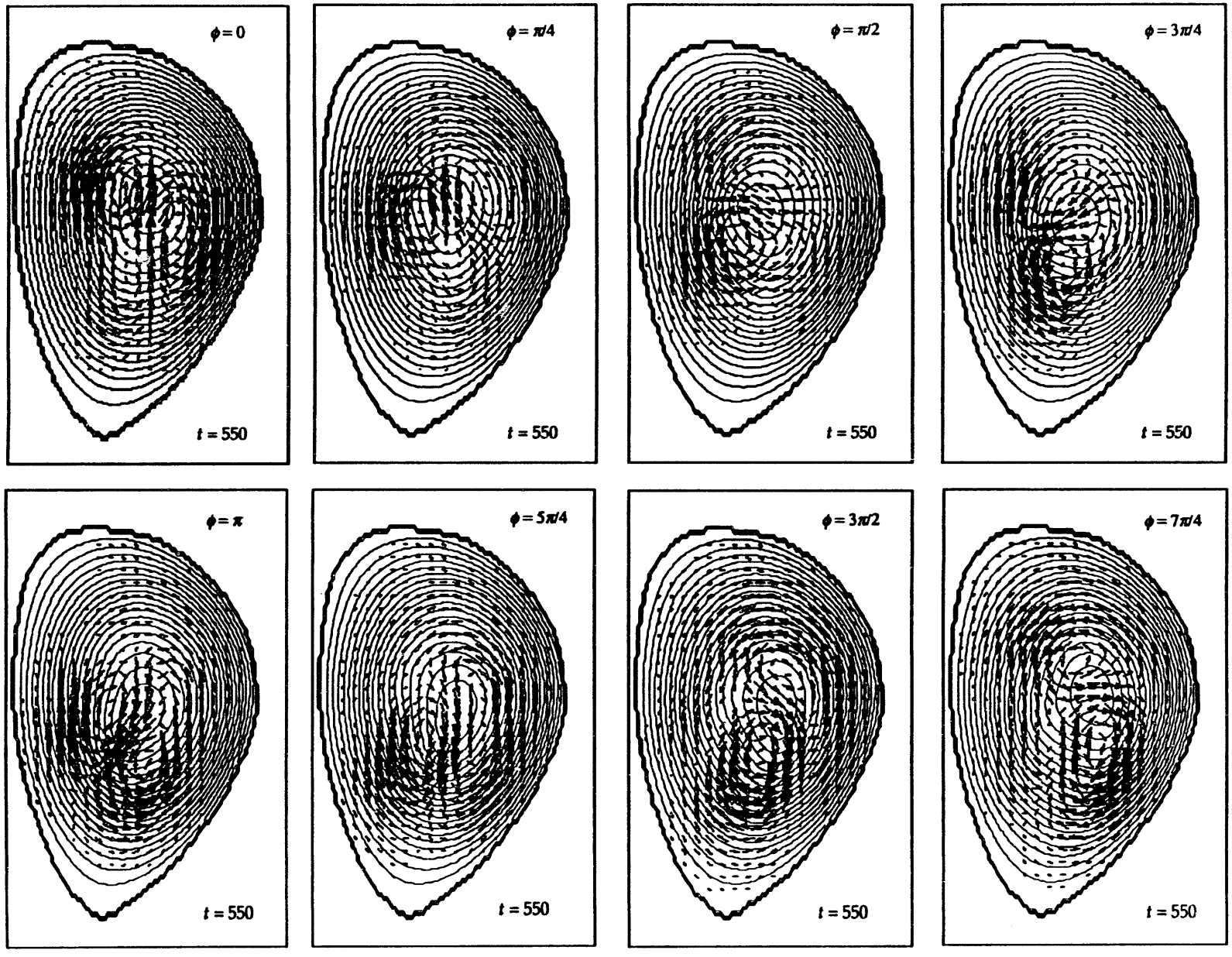


Figure 6-21. Velocity and poloidal flux at 8 different toroidal locations late in the linear growth phase.

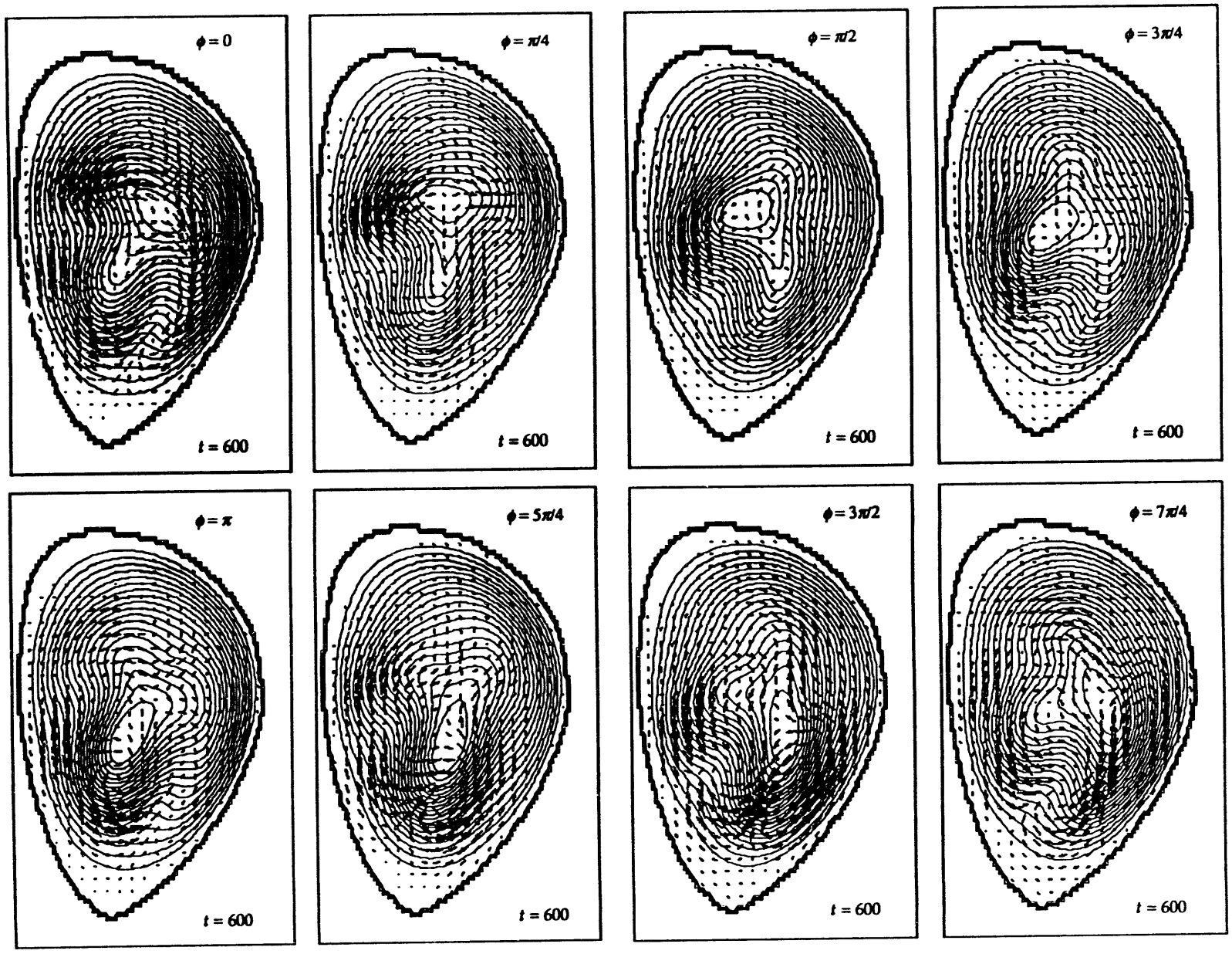


Figure 6-22. Velocity and poloidal flux at 8 different toroidal locations at the time of nonlinear saturation.

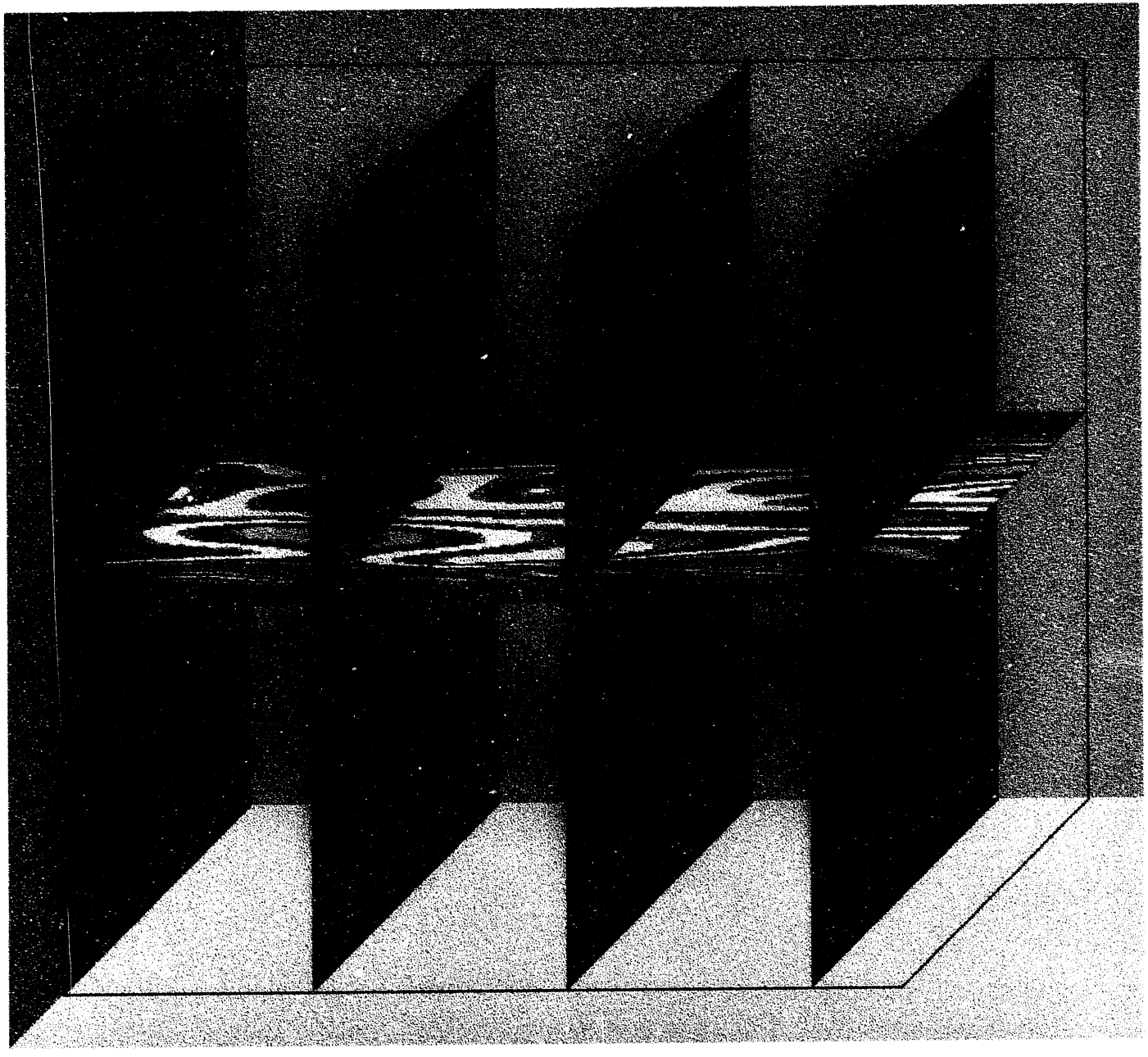


Figure 6-23. Three dimensional structure of the pressure at the time of nonlinear saturation.

7.0 SUMMARY AND CONDITIONS

An algorithm for the solution of the time-dependent, primitive, resistive MHD equations in three-dimensional toroidal geometry has been developed. The algorithm uses an unstructured, triangular mesh in the poloidal plane, and a structured, pseudospectral method based on FFTs in the toroidal coordinate. The algorithm is fully conservative and maintains both the magnetic field and current density as solenoidal. Fluxes at cell interfaces are computed with a low-order upwind method. The semi-implicit method is used for time integration.

A code based on the algorithm, TRIM, has been written and is being verified. The first order nature of the algorithm has been confirmed. The code has been applied to four nonlinear test problems: a hydrodynamic shock tube; an MHD shock tube; toroidal force balance; and, growth and saturation of toroidal instabilities. For both the hydrodynamic and MHD shock tube problems, good agreement with previous results has been obtained. The primary inaccuracy is due to the numerical diffusion introduced by the low order fluxes. Mesh adaption and refinement has been successfully applied in the hydrodynamic case. Toroidal force balance has been computed by viscous damping of Alfvén and sound waves. Linear growth and nonlinear saturation of a three-dimensional kink mode in a highly elongated toroidal equilibrium representative of the ITER design has been computed. This calculation is qualitatively correct, but quantitative differences with other calculations must still be resolved. These differences may be due to resistive profile modification, nonlinear effects, or errors.

During the next year we will concentrate on verification of results and algorithm improvement. In particular,

We will obtain an understanding and confirmation of the toroidal stability results described in Section 6.4,

We will develop techniques to apply the mesh adaption algorithm to MHD,

We will incorporate implicit resistivity,

We will investigate methods of interpolation on unstructured meshes,

If necessary, we will investigate higher order methods for determining interface fluxes,

We will begin to apply the algorithm to the external kink mode,

We will investigate the viability of applying the algorithm to dynamics and transport in the tokamak scrape-off layer (SOL), and

We will visit various national fusion facilities to publicize the algorithm and to offer its use in solving fusion problems.

REFERENCES

1. S. Migliuolo, *Nucl. Fusion* **33**, 1721 (1993).

2. Liu Chen *et al.*, *Phys. Rev. Lett.* **52**, 1122 (1984); B. Coppi and F. Porcelli, *Phys. Rev. Lett.* **57**, 2272 (1986); B. Coppi *et al.*, *Phys. Fluids* **31**, 1630 (1988).

3. B. A. Carraras *et al.*, *Phys. Fluids* **23**, 1811 (1980).

4. D. C. Barnes, private communication.

5. S. Ortolani and D. D. Schnack, *Magnetohydrodynamics of Plasma Relaxation*, World Scientific Press, Singapore, 1993.

6. For an excellent introduction and review, see the articles contained in *Unstructured Grid Methods for Advection Dominated Flows*, AGARD R-787, Special Course Notes presented at the von Kármán Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, 2-6 March 1992 and at the NASA Ames Research Center, Moffett Field, CA, 28 Sept.-2 Oct. 1992, Advanced Group for Aerospace Research and Development (AGARD), 7 Rue Ancelle 92200 Neuilly-sur-Seine, France, 1992.

7. R. Courant and K. O. Friedrichs, *Supersonic Flows and Shock Waves*, Interscience Publishers, Inc., New York, 1948.

8. D. S. Harned and W. Kerner, *J. Comp. Phys.* **60**, 62 (1985); D. S. Harned and W. Kerner, *Nucl. Sci. Eng.* **92**, 119 (1986); D. S. Harned and D. D. Schnack, *J. Comp. Phys.* **65**, 57 (1986).

9. D. Nielson, private communication.

10. D. D. Schnack *et al.*, *J. Comp. Phys.* **70**, 330 (1987).

11. T. J. Barth, "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations", in Ref. 6, above.

12. P. J. Green and R. Sibson, *The Computer Journal* **21**, 191 (1983).

13. R. D. Rausch, "Time Marching Aeroelastic and Spatial Adaption Procedures on Triangular and Tetrahedral Meshes Using an Unstructured Grid Euler Method", Ph. D. Thesis, Purdue University, December 1992.

14. G. Sod, *J. Comp. Phys.* **27**, 1 (1978).

15. M. Brio and C. C. Wu, *J. Comp. Phys.* **75**, 400 (1988).

16. L. S. Solov'ev, *Sov. Physics JETP* **26**, 400 (1968).

17. A. Turnbull, private communication (1994).

18. R. Gruber *et al., Comput. Phys. Commun.* **24**, 363 (1981).

DATE
FILMED
10/18/94

END