DOE/ER/54124--T1

SAIC-95/1323:APPAT-170
September 15, 1995

TRANSPORT AND DYNAMICS IN TOROIDAL FUSION SYSTEMS

FINAL REPORT
1992-1995

Science Applications International Corporation
An Employee-Owned Company

Worked Performed Under
DOE Grant DE-FG03-91ER54124

DALTON D. SCHNACK
PRINCIPAL INVESTIGATOR
Applied Physics Operation
Science Applications International Corp.
San Diego, CA 92121-1578

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

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, whose geometry is 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. 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 summary and discussion 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 $\mathbf{r}_i$, 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.

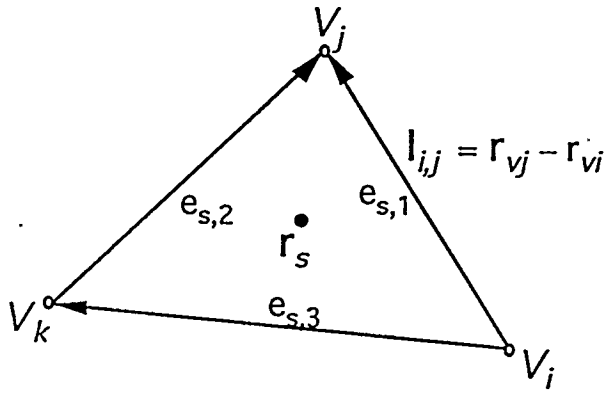

Figure 2-1. Triangle, edge, and vertex mesh elements.

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[9] for use in magnetohydrodynamics. These routines interface with unstructured meshes generated by the SMART code[10]. 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.

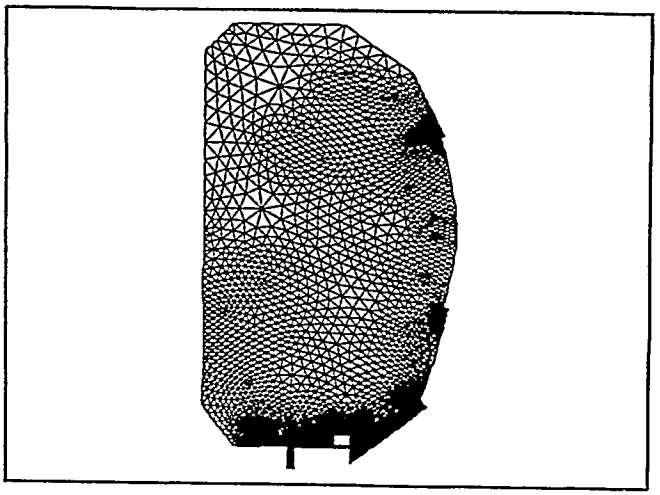

Figure 2-2. An example of an unstructured mesh describing the geometry of the D-III-D experiment.

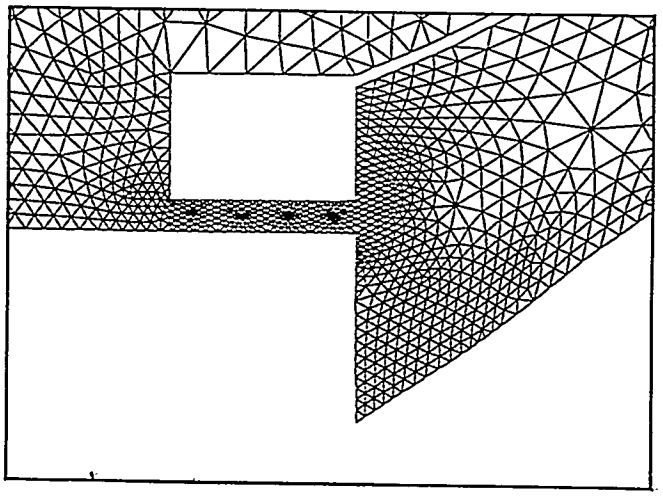

Figure 2-3. Details of the zoning in the divertor region of D-III-D (the bottom portion of Figure 2-2).

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[11].

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.

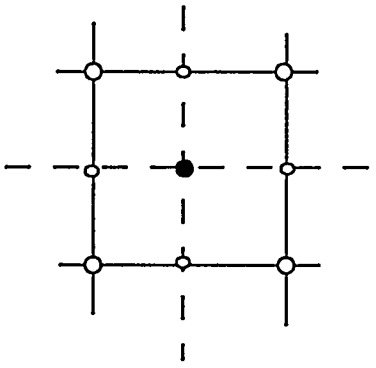

Figure 2-4. Structured, staggered (dual) meshes.

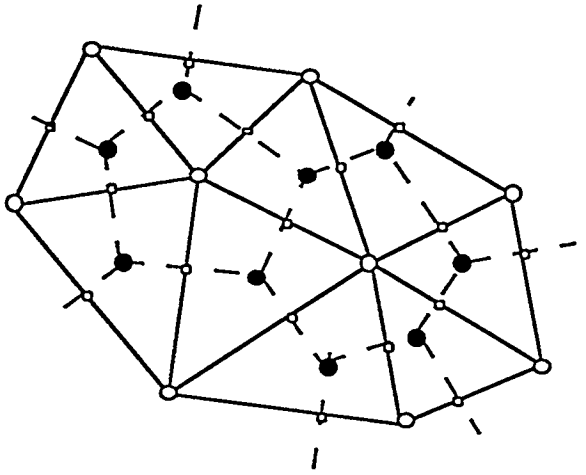

Figure 2-5. Triangular (primary) and polygon (dual) meshes.

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. 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[12] 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. Because of this, 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 calculation of some dependent variables, as will be discussed in Section 4.

2.6 The Third (Toroidal) Dimension

Our goal is to describe magnetohydrodynamics in toroidal fusion systems. The geometry of these systems is symmetric about an axis, and is best described in cylindrical (r,ϕ,z) coordinates. We use the unstructured Delaunay triangular mesh and the barycenter dual mesh dual to approximate the geometry in the poloidal (r, z) plane. Since the angular (ϕ) coordinate is periodic, and since the dominant MHD motions in a tokamak are long wavelength in this direction, we have chosen a pseudospectral description using Fast Fourier Transforms (FFTs) for this coordinate. 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.

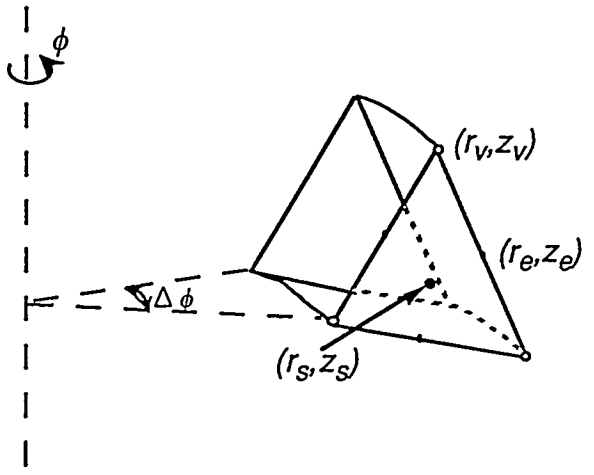

Figure 2-6. Three-dimensional control volume.

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[13] 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. Triangle deletion is sketched in Figure 2-8. 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 or coarsening. We have found a modified version of the classic interpolation estimate originally developed for steady-state hydrodynamic computations[14,15] to be useful. For each triangle s, we compute the normalized second derivative

$$E_S = \frac{\delta_S{}^2\overline{|\nabla^2 U_S|}}{\delta_S\overline{|\nabla U_S|}+\varepsilon\overline{|U_S|}}, \tag{2.2}$$

where $\delta_S = \Delta a_S{}^{1/2}$, U_S is any triangle-centered dependent variable, ε is a constant between 0 and 1, and the overscore indicates an average over triangle s and its three neighbors. (For three-dimensional problems, the maximum of E_S over the toroidal dimension is taken.) The quantity E_S is dimensionless and bounded, so that it can be used for a variety of problems and dependent variables. All triangles for which $E_S > E_R$ are refined, while all triangles for which $E_S < E_C$ are coarsened. Typically we have used the values $\varepsilon = 0.2$, $E_R = 0.8$ and $E_C = 0.2$.

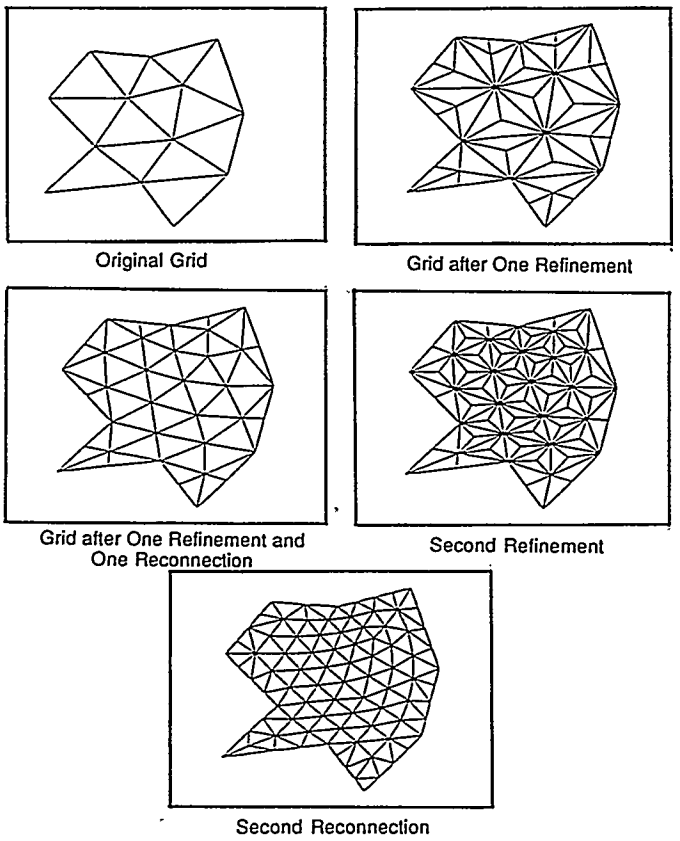

Figure 2-7. Mesh refinement and edge swapping (from Ref. 9)

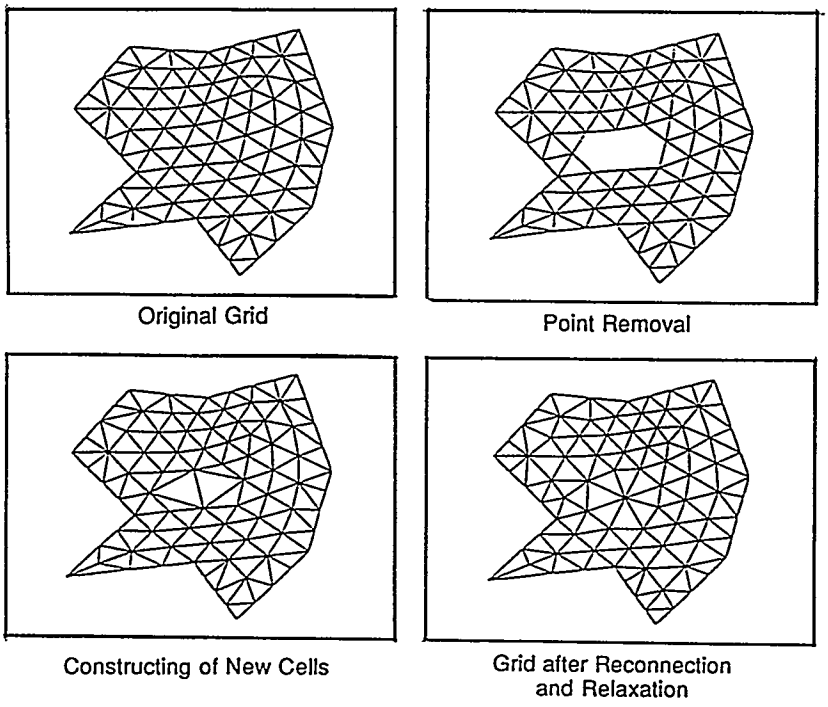

Figure 2-8. Mesh coarsening (from Ref. 9).

As mesh refinement and coarsening are time consuming operations they are not done every timestep. Mesh refinement is done after every N_R timesteps, and coarsening is done after every N_C refinement steps. Values of N_R from 5 to 20, and N_C from 2 to 4, are typical. Examples of refinement and coarsening are given in Section 6.

3.0 APPROXIMATION TO SPATIAL DIFFERENTIAL OPERATORS

3.1 The Resistive MHD Equations

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

$$\frac{\partial \mathbf{A}}{\partial t} = -\mathbf{E} \tag{3.1}$$

$$\mathbf{E} = -\mathbf{v}\times\mathbf{B}+\eta\mathbf{J}/S \tag{3.2}$$

$$\mathbf{B} = \nabla\times\mathbf{A} \tag{3.3}$$

$$\mathbf{J} = \nabla \times \mathbf{B} \tag{3.4}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} = -\nabla \cdot \mathbf{T} \tag{3.5}$$

$$\mathbf{T} = \rho \mathbf{v}\mathbf{v} - \mathbf{B}\mathbf{B} + \frac{1}{2}\left(p + B^2\right)\mathbf{I} \tag{3.6}$$

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}) \tag{3.7}$$

$$\frac{\partial u}{\partial t} = -\nabla \cdot \mathbf{F} \tag{3.8}$$

$$u = \rho v^2 + B^2 + \frac{p}{\gamma - 1} \tag{3.9}$$

$$\mathbf{F} = \left(\rho v^2 + \frac{\gamma}{\gamma - 1} p\right)\mathbf{v} + 2\, \mathbf{E} \times \mathbf{B} \tag{3.10}$$

where γ is the ratio of specific heats, u is the total energy density, F is the energy flux, I is the unit tensor, and 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.

3.2 The Finite Volume Approximation

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

$$\mathbf{e}_\phi = \mathbf{t}_e \times \mathbf{n}_e \tag{3.11}$$

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

$$\mathbf{l}_e = \Delta r_e\, \mathbf{e}_r + \Delta z_e\, \mathbf{e}_z = \Delta l_e \mathbf{t}_e \tag{3.12}$$

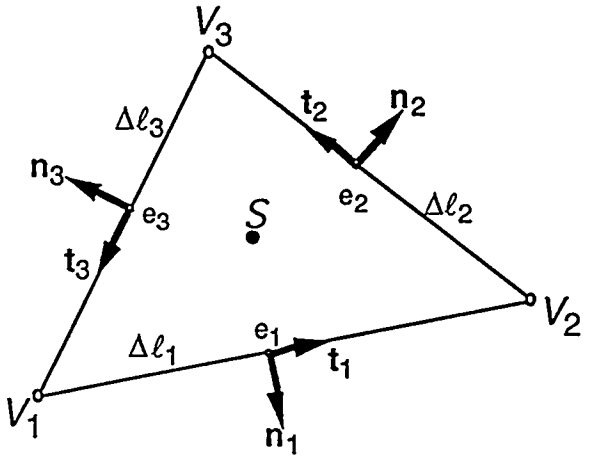

Figure 3-1. Poloidal projection of control volume.

$$\mathbf{n}_e = \frac{\Delta z_e\, \mathbf{e}_r - \Delta r_e\, \mathbf{e}_z}{\Delta l_e} \tag{3.13}$$

and the area of the triangle is

$$\Delta a_s = \frac{1}{2}\left|\mathbf{l}_1 \times \mathbf{l}_2\right| = \frac{1}{2}\left|\mathbf{l}_2 \times \mathbf{l}_3\right| = \frac{1}{2}\left|\mathbf{l}_3 \times \mathbf{l}_1\right| . \tag{3.14}$$

The unit normal vector $\mathbf{n}_e$ points from the left side of edge e (triangle $s = L_e$) to the right side of edge e (triangle $s = R_e$).

The finite volume method is used 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

$$f(r,\phi,z,t) = \sum_n f_n(r,z,t)e^{in\phi} \tag{3.15}$$

and then integrate the appropriate identity over the three-dimensional control volume shown in Figure 2-6. (Since the toroidal representation is spectral and not finite-difference, the limit of the resulting expressions as $\Delta\phi \to 0$ is taken.) 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 (3.15) into the integral identity

$$\int \nabla f dV = \oint f\mathbf{n} dS , \tag{3.16}$$

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

$$(\nabla f)_s = \frac{1}{r_s \Delta a_s} \sum_{e=1}^{3} r_e \Delta l_e \mathbf{n}_e \bar{f}_e - \frac{f_s}{r_s} \mathbf{e}_r + \frac{in}{r_s} f_s \mathbf{e}_\phi \ . \tag{3.17}$$

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. The quantity $\bar{f}_e$ is the simple average $\bar{f}_e = (f_{Re} + f_{Le})/2$, where the values f_{Re} and f_{Le} are the values of f_s in the triangles lying to the right (R_e) and left (L_e) of edge e.

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

$$\int \nabla \cdot \mathbf{A} dV = \oint f\mathbf{n} \cdot \mathbf{A} dS \tag{3.18}$$

to obtain the approximation

$$(\nabla \cdot \mathbf{A})_s = \frac{1}{r_s \Delta a_s} \sum_{e=1}^{3} r_e \Delta l_e \mathbf{n}_e \cdot \mathbf{A}_e + \frac{in}{r_s} A_{\phi s} \ ; \tag{3.19}$$

for the *curl of a vector* we use

$$\int \nabla \times \mathbf{A} dS = \oint \mathbf{t} \cdot \mathbf{A} dl \tag{3.20}$$

to obtain the approximation

$$(\nabla \times \mathbf{A})_{ne} = -\frac{1}{r_e \Delta l_e} \left(r_{v+} A_{\phi v+} - r_{v-} A_{\phi v-} \right) + \frac{in}{r_e} \mathbf{A}_e \cdot \mathbf{t}_e \ , \tag{3.21}$$

$$(\nabla \times \mathbf{A})_{\phi s} = -\frac{1}{\Delta a_s} \sum_{e=1}^{3} \mathbf{A}_e \cdot \mathbf{l}_e \ . \tag{3.22}$$

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

$$\int \nabla \cdot \mathbf{T} dV = \oint f\mathbf{n} \cdot \mathbf{T} dS \tag{3.23}$$

to obtain

$$(\nabla\cdot\mathrm{T})_s = \frac{1}{r_s\Delta a_s}\sum_{e=1}^{3} r_e\Delta l_e\left[\mathbf{e}_r(\mathbf{n}_e\cdot\mathrm{T}_e)_r + \mathbf{e}_\phi(\mathbf{n}_e\cdot\mathrm{T}_e)_\phi + \mathbf{e}_z(\mathbf{n}_e\cdot\mathrm{T}_e)_z\right]$$

$$+ \frac{in}{r_s}\left[\mathbf{e}_r T_{\phi rs} + \mathbf{e}_\phi T_{\phi\phi s} + \mathbf{e}_z T_{\phi zs}\right] + \frac{1}{r_s}\left[\mathbf{e}_\phi T_{\phi rs} - \mathbf{e}_r T_{\phi\phi s}\right] . \tag{3.24}$$

It is easy to verify from Equations (3.19), (3.21), and (3.22) 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

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. (The quantities $\rho_s\Delta V_s$, $(\rho\mathbf{v})_s\Delta V_s$, and $u_s\Delta V_s$ are the total mass, momentum, and energy in cell s.) 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 mass density in triangle s is given by

$$\frac{\partial\rho_s}{\partial t} = -\frac{1}{r_s\Delta a_s}\sum_{e=1}^{3} r_e\Delta l_e\mathbf{n}_e\cdot\mathrm{F}_e + \frac{in}{r_s}F_{\phi s} \tag{4.1}$$

where

$$\mathbf{n}_e\cdot\mathrm{F}_e = \rho_e\, v_{ne} \tag{4.2}$$

is the poloidal mass flux across edge e, and

$$F_{\phi s} = \rho_s\, v_{\phi s} \tag{4.3}$$

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

$$v_{ne} = \frac{1}{2}\,\mathbf{n}_e\cdot(\mathbf{v}_{Le}+\mathbf{v}_{Re}) . \tag{4.4}$$

Expressions similar to Equations (4.1–4.3) hold for the momentum equation (Eq. 3.5), and the energy equation (Eq. 3.8).

The advective flux at an edge e is computed using the full donor cell method. For example, the right hand side of Equation (4.2) is evaluated as

$$(F_{ne})_{adv} = \rho_{Le}\, v_{ne}\,, \quad \text{if} \quad v_{ne} > 0 \tag{4.5a}$$

$$(F_{ne})_{adv} = \rho_{Re}\, v_{ne}\,, \quad \text{if} \quad v_{ne} < 0 \tag{4.5b}$$

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 $v_{ne}\delta_s/2$, where $\delta_s \approx \Delta a_s{}^{1/2}$. While this technique may be too diffusive for highly accurate shock calculations, it is quite adequate to describe the relatively slow motions of interest in tokamak dynamics. Problems involving strong shocks may 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 (3.21) and (3.22) 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 $(\nabla \cdot \mathrm{B})_s \equiv 0$.)

The procedure described above defines only the component of B normal to each edge. 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 (3.20) over the surface of the dual polygon p_e that crosses an edge e, as shown in Figure 4-1.

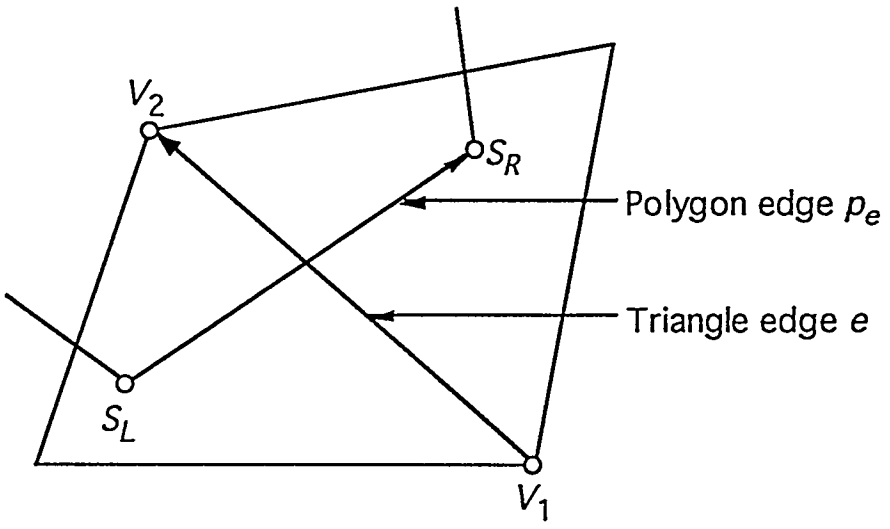

Figure 4-1. Triangle and polygon edges.

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 at a triangle edge are then given by

$$B_{re} = \frac{1}{\Delta}\left(B_{ne}n_{pez} - B_{npe}n_{ez}\right) \tag{4.6}$$

$$B_{ze} = \frac{1}{\Delta}\left(B_{npe}n_{er} - B_{ne}n_{per}\right) \tag{4.7}$$

where

$$\Delta = \mathbf{e}_{\phi} \cdot \left(\mathbf{n}_e \times \mathbf{n}_{pe}\right) \neq 0 , \tag{4.8}$$

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

$$B_{te} = B_{ze}n_{er} - B_{re}n_{ez} . \tag{4.9}$$

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 (3.1), 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

$$E_{ne} = -\,\overline{v}_{\phi e}B_{te} + \overline{v}_{ne}\overline{B}_{te} + \eta J_{ne}/S , \tag{4.10}$$

$$E_{te} = -\,\overline{v}_{ne}\overline{B}_{\phi e} + \overline{v}_{\phi e}B_{ne} + \eta J_{te}/S . \tag{4.11}$$

The toroidal electric field at the triangle centroids is given by

$$E_{\phi s} = -\,v_{zs}\overline{B}_{rs} + v_{rs}\overline{B}_{zs} + \eta J_{\phi s}/S . \tag{4.12}$$

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:

$$\bar{f}_e = \frac{1}{2}\left(f_{Re} + f_{Le}\right) . \tag{4.13}$$

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

$$\bar{f}_s = \frac{1}{3}\left(f_{v1} + f_{v2} + f_{v3}\right) , \tag{4.14}$$

and for functions defined at edges

$$\bar{f}_s = \frac{1}{3}\left(f_{e1} + f_{e2} + f_{e3}\right) , \tag{4.15}$$

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[16]. In this approach, the interpolated value of a function at vertex v is given by the weighted average

$$\bar{f}_v = \frac{\sum_{s'}(1+w_{s'})f_{s'}}{\sum_{s'}(1+w_{s'})} , \tag{4.16}$$

where the prime (s') indicates that the sums are taken over all triangles sharing vertex v, and *not* over all triangles N_s.

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

$$F(w_s) = \sum_{s'} w_{s'}^2 \tag{4.17}$$

subject to the constraints

$$L_r = \sum_{s'}(1+w_{s'})(r_{s'} - r_v) = 0 , \tag{4.18a}$$

$$L_r = \sum_{s'}(1+w_{s'})(r_{s'} - r_v) = 0 , \tag{4.18b}$$

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

$$w_s = \lambda_r \Delta r_s + \lambda_z \Delta z_s \ , \tag{4.19}$$

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

$$\lambda_r = \frac{R_z I_{rz} - R_r I_{zz}}{I_{rr} I_{zz} - I_{rz}^2} \ , \tag{4.20}$$

$$\lambda_z = \frac{R_z I_{rz} - R_z I_{rr}}{I_{rr} I_{zz} - I_{rz}^2} \ , \tag{4.21}$$

$$R_r = \sum_{s'} \Delta r_{s'} \ , \tag{4.22}$$

$$R_z = \sum_{s'} \Delta z_{s'} \ , \tag{4.23}$$

$$I_{rr} = \sum_{s'} (\Delta r_{s'})^2 \ , \tag{4.24}$$

$$I_{zz} = \sum_{s'} (\Delta z_{s'})^2 \ , \tag{4.25}$$

$$I_{rz} = \sum_{s'} \Delta r_{s'} \Delta z_{s'} \ . \tag{4.26}$$

Equations (4.13–4.16) are first order accurate, i.e., they are exact for linear functions. (Equations (4.14) and (4.15) are just special cases of Equation (4.16)). The use of higher order interpolation methods, especially in place of Equation (4.13), can be shown to lead to a non-Hermitian formulation and resulting unphysical behavior.

A complication is that neither Equations (4.13) and (4.15), nor Equations (4.14) and (4.16), are exact inverses of each other. Thus, for example, interpolation from centroids to vertices using Equation (4.16), followed directly by interpolation from vertices to centroids using Equation (4.14), introduces errors. Heuristically, these errors do not seem critical to the results obtained with the algorithm, but their affect on problems in parameter regimes other than those studied in the present work cannot be assessed.

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.

Boundary conditions corresponding to $r = 0$, to applied tangential or toroidal electric fields, and to "simple" inflow (specified upstream pressure and density, zero normal derivative of velocity) and outflow (zero normal derivative of pressure, density, and velocity) have also been implemented.

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.,

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

or,

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

$$\frac{U^{**} - U^*}{\Delta t} = F_{\text{semi-implicit}} \tag{5.2b}$$

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

where U is the state-vector describing the system, and $\mathbf{F}_{explicit}$, $\mathbf{F}_{semi\text{-}implicit}$, and $\mathbf{F}_{viscous}$ 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:

$$\frac{\rho \mathbf{v}^* - \rho \mathbf{v}^{n-1/2}}{\Delta t} = -\nabla \cdot (\rho \mathbf{v}\mathbf{v})^{n-1/2} \tag{5.3}$$

$$\frac{\rho \mathbf{v}^{n+1/2} - \rho \mathbf{v}^{n-1/2}}{\Delta t} = -\nabla \cdot \left[(\rho \mathbf{v}\mathbf{v})^* - (\mathbf{B}\mathbf{B})^n \right] - \frac{1}{2} \nabla \left(p + B^2 \right)^n \tag{5.4}$$

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

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

$$\mathbf{B}^* = \nabla \times \mathbf{A}^* \tag{5.7}$$

$$\frac{\mathbf{A}^{n+1} - \mathbf{A}^n}{\Delta t} = \mathbf{v}^{n+1/2} \times \mathbf{B}^* - \eta \mathbf{J}^n / S \tag{5.8}$$

$$\frac{\rho^* - \rho^n}{\Delta t} = -\nabla \cdot \left(\rho^n \mathbf{v}^{n+1/2} \right) \tag{5.9}$$

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

$$\frac{u^* - u^n}{\Delta t} = -\nabla \cdot \frac{\gamma}{\gamma - 1} p^n \mathbf{v}^{n+1/2} \tag{5.11}$$

$$p^* = (\gamma-1)\left[u^* - \left(\rho v^2\right)^{n+1/2} - \left(B^2\right)^{n+1}\right] \tag{5.12}$$

$$\mathbf{E}^{n+1} = -\mathbf{v}^{n+1/2}\times\mathbf{B}^{n+1} + \eta\mathbf{J}^{n+1}/S \tag{5.13}$$

$$\frac{u^{n+1}-u^n}{\Delta t} = -\nabla\cdot\left\{\left[\left(\rho v^2\right)^{n+1/2} + \frac{\gamma}{\gamma-1}\,p^*\right]\mathbf{v}^{n+1/2} + 2\,\mathbf{E}^{n+1}\times\mathbf{B}^{n+1}\right\} \tag{5.14}$$

$$p^{n+1} = (\gamma-1)\left[u^{n+1} - \left(\rho v^2\right)^{n+1/2} - \left(B^2\right)^{n+1}\right] \tag{5.15}$$

Total mass, momentum, and magnetic flux are exactly conserved. Because the pressure, magnetic field, mass density, and momentum are defined at different time levels, the sum of the kinetic, magnetic, and internal energies is exactly conserved in the limit $\Delta t \to 0$, independent of spatial discretization. (The volume integral of the quantity u is exactly conserved independent of Δt.) 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 by 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[11]. 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

$$\left(1-\alpha\Delta t\nabla^2\right)(\rho\mathbf{v})^{**} = (\rho\mathbf{v})^* - \alpha\Delta t\nabla^2(\rho\mathbf{v})^n\,, \tag{5.16}$$

where α is the semi-implicit coefficient given by

$$\alpha = \frac{\sigma}{k_{\max}^2 \Delta t}\left[\left(\frac{\Delta t}{\Delta t_{\mathrm{CFL}}}\right)^2 - 1\right] \text{ for } \Delta t > \Delta t_{\mathrm{CFL}}\,, \tag{5.17}$$

$$\alpha = 0 \text{ for } \Delta t \leq \Delta t_{\mathrm{CFL}}\,, \tag{5.18}$$

where $(\rho \mathbf{v})^*$ is the value of momentum obtained from the explicit advance (Eq. (5.4)), $k_{\max} \approx 1/\delta_s$ is the largest wave-number resolved on the mesh, Δt_{CFL} is the maximum time step allowed by the CFL restriction for normal modes, and σ is a constant $\gtrsim 1$.

The time step is completed with the implicit viscous advance

$$\left(1 - \nu \Delta t \nabla^2\right)(\rho \mathbf{v})^{n+1} = (\rho \mathbf{v})^{**}\,, \tag{5.19}$$

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

The vector Laplacian operator appearing in Equations (5.16) and (5.19) requires the definition of the scalar Laplacian. This is accomplished by the successive application of the gradient and divergence operators defined in Equations (3.17) and (3.19). When combined with the boundary condition

$$(\nabla f)_G = (\nabla f)_B\,, \tag{5.20}$$

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.

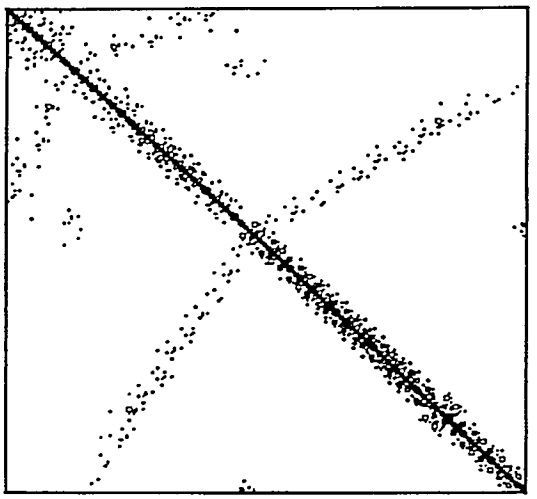

Figure 5-1. Matrix structure for the Laplacian operator on an unstructured mesh.

Matrix inversion is performed with a conjugate gradient (CG) algorithm with diagonal preconditioning[17]. 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.

At this time the resistivity is treated explicitly. Since $\eta/S << 1$ we have not found this to be computationally restrictive.

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[18]. 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 initial 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. It contains 1250 triangles. In this figure, the initial membrane is horizontal, and centered at $z = 0.5$. As the solution proceeds in time no spatial variation develops in the direction parallel to the membrane. The solution thus remains one-dimensional, even with the two-dimensional algorithm. The analytic solution at $t = 0.1$ is shown in Figure 6-3a-c.

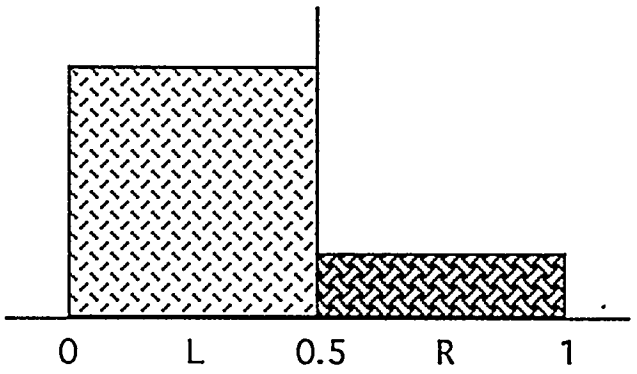

Figure 6-1. Initial conditions for hydrodynamic shock tube problem.

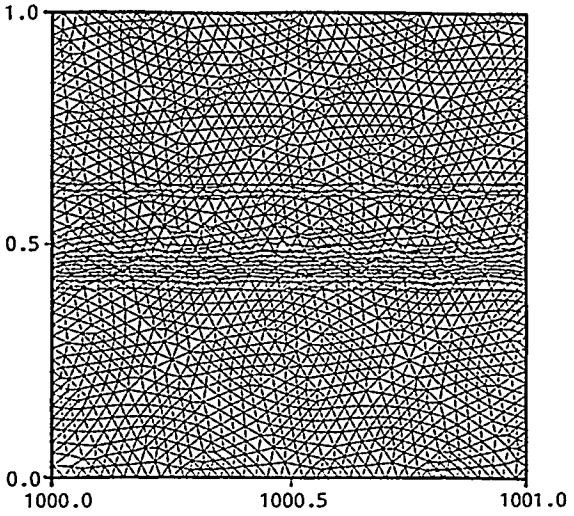

Figure 6-2. Mesh for hydrodynamic shock tube problem, with superimposed contours.

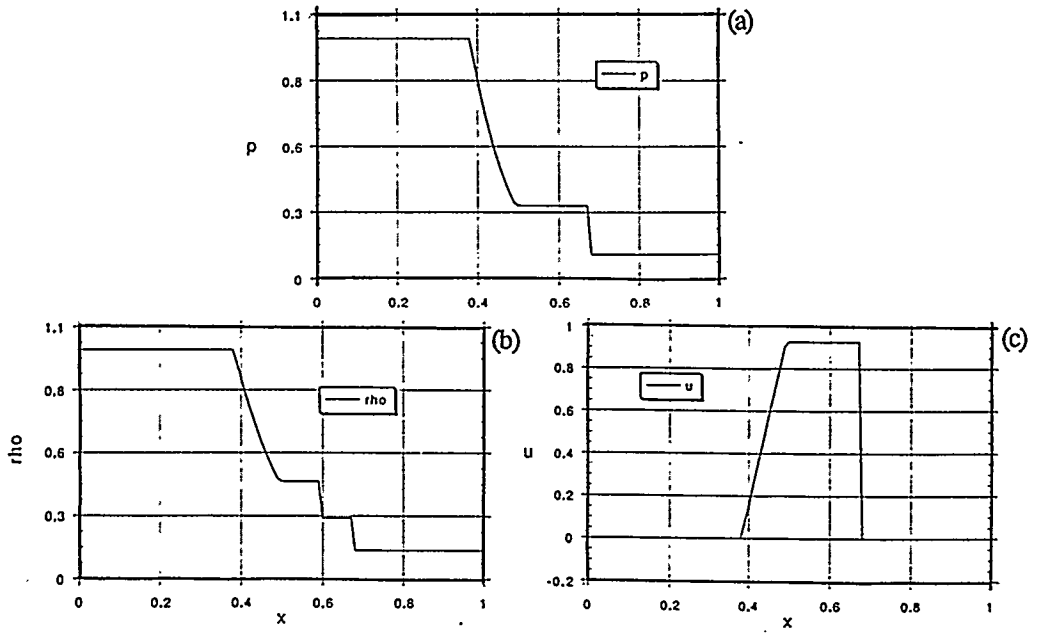

Figure 6-3(a-c). Analytic solution to the hydrodynamic shock problem.

The results of TRIM with the mesh shown in Figure 6-2 is shown in Figure 6-4(a-c) at $t = 0.1$. 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.

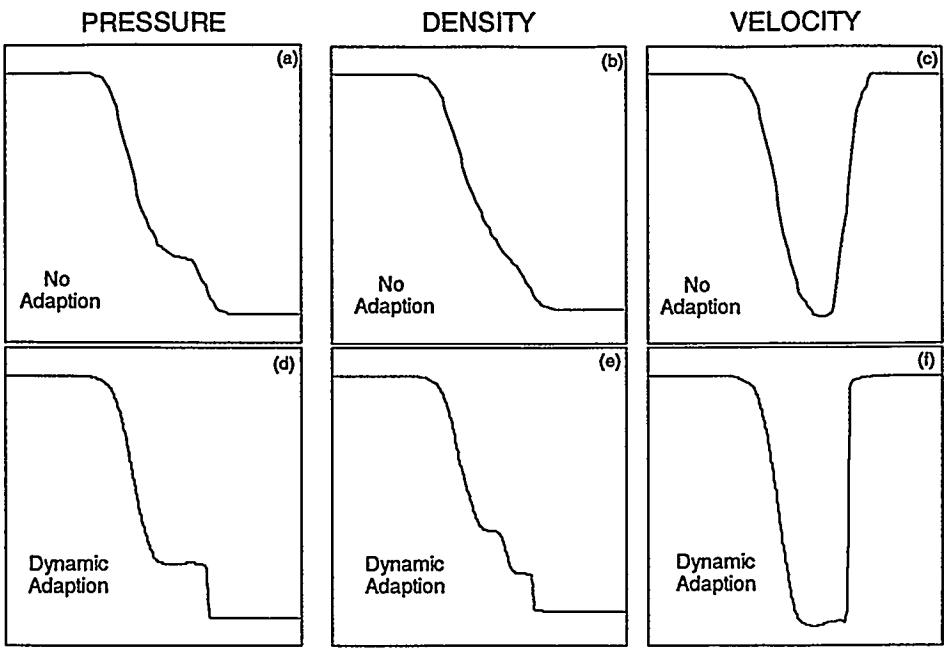

Figure 6-4(a-f). Comparison of numerical solutions without (a-c) and with (d-f) dynamic mesh adaption.

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. For this problem we used the mass density ρ as the dependent variable in the refinement criterion, Eq. (2.2), with $\varepsilon = 0.2$, $E_R = 0.8$ and $E_C = 0.2$. Both refinement and coarsening were done every 50 timesteps. The initial mesh is shown in Figure 6-5a. This mesh has been refined in order to initially capture the pressure and density discontinuities. The adaptively refined mesh at $t = 0.1$ is shown in Figure 6-5b. The algorithm has adapted the mesh to the dynamically evolving shock, contact discontinuity and expansion front. The initial mesh had 7777 triangles, and the dynamically evolving mesh contained up to 34415 triangles. In Figure 6-4(a-f) we compare the solution at $t = 0.1$ with and without adaption. All features are sharper with dynamic mesh refinement than without.

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.

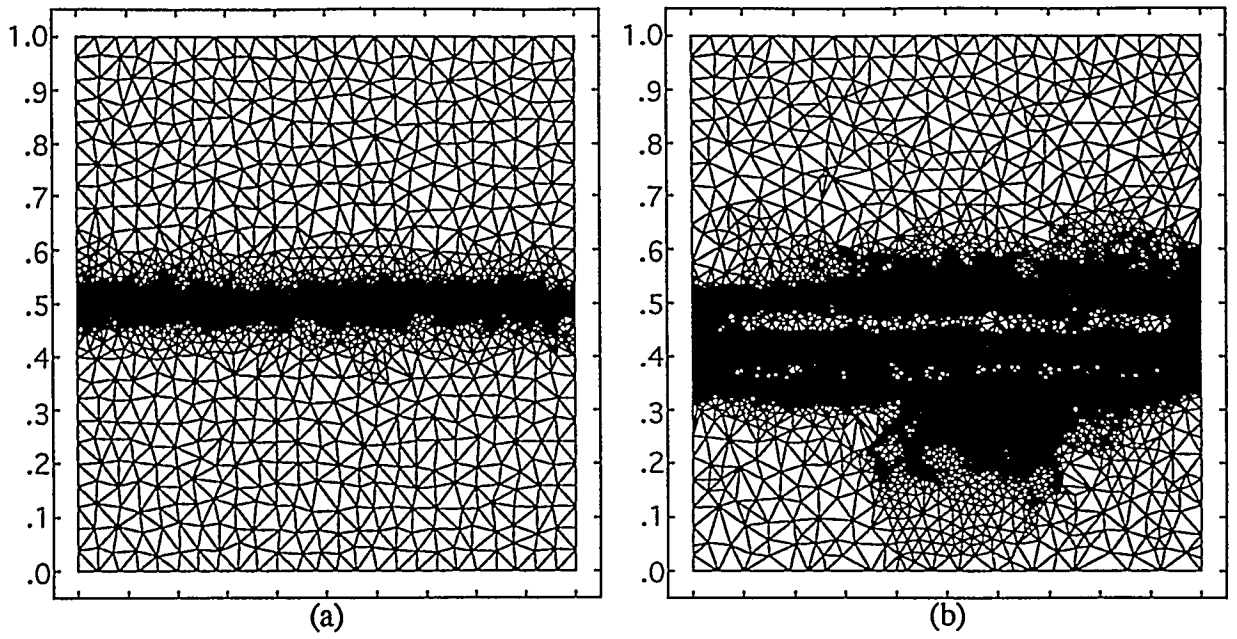

Figure 6-5. (a) Initial mesh for dynamic adaption; (b) Dynamically adapted mesh at $t = 0.1$.

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[19]. 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.

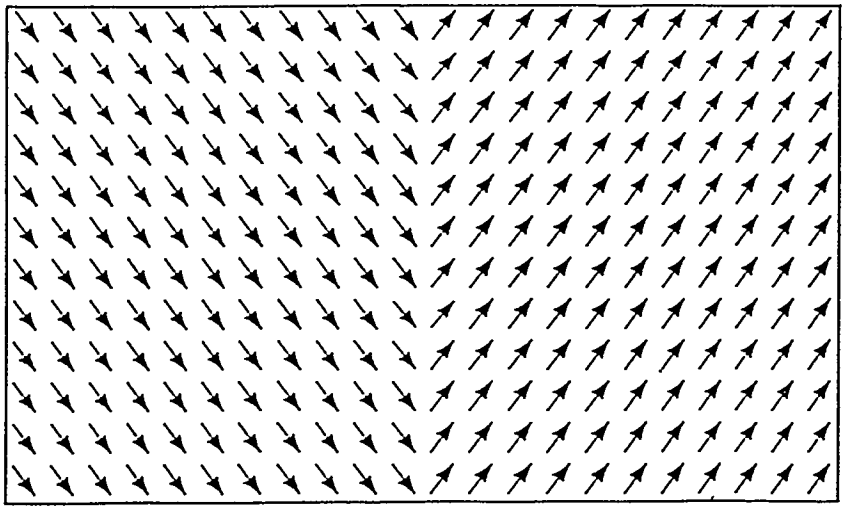

Figure 6-6. Initial magnetic field vectors for the MHD shock tube problem.

The dynamics after the membrane is ruptured are much more complex than in the purely hydrodynamic case; we refer the reader to Ref. 19 for details. In Figure 6-7 we present our two-dimensional solution of this problem. This can be compared with the more finely resolved one-dimensional solution of Brio and Wu[19]. 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.

We have also applied mesh refinement and coarsening to the MHD shock tube problem. The refinement and coarsening criteria are the same as described in Section 6.1. Figure 6-8 shows a comparison of the mass density ρ with and without dynamic mesh adaption. Finer structure is observed when dynamic mesh refinement is implemented. However, the contact discontinuity is still poorly resolved due to the low order calculation of the interface fluxes.

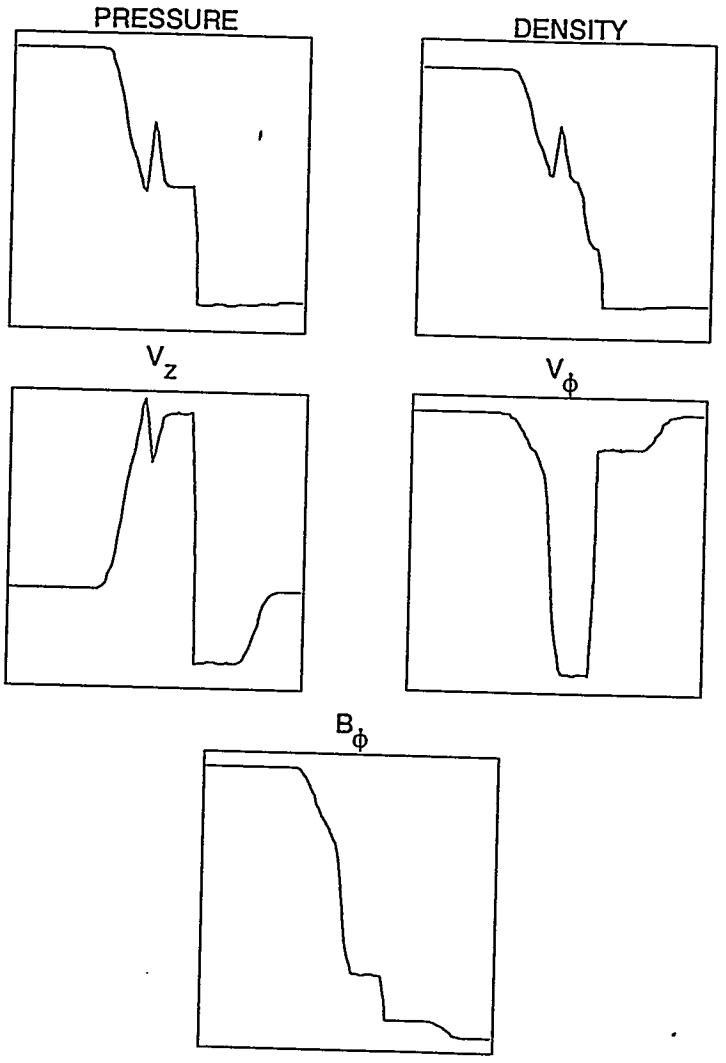

Figure 6-7. Numerical solution of the MHD shock tube problem. This can be compared with the solution given in Ref. 19.

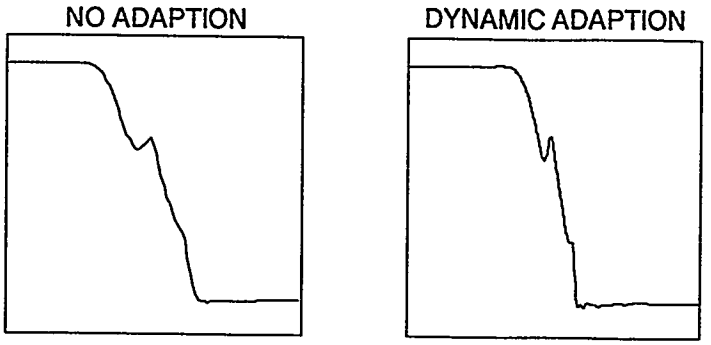

Figure 6-8. Comparison of the results for the MHD shock tube problem without and with dynamic mesh adaption.

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[20]. 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 6-9. Since $b = 0$, this equilibrium has no poloidal current ($J_r = J_z = 0$).

A typical unstructured mesh for this problem with $N_s = 1658$ is shown in Figure 6-10. 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 (3.5) (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 6-11 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.

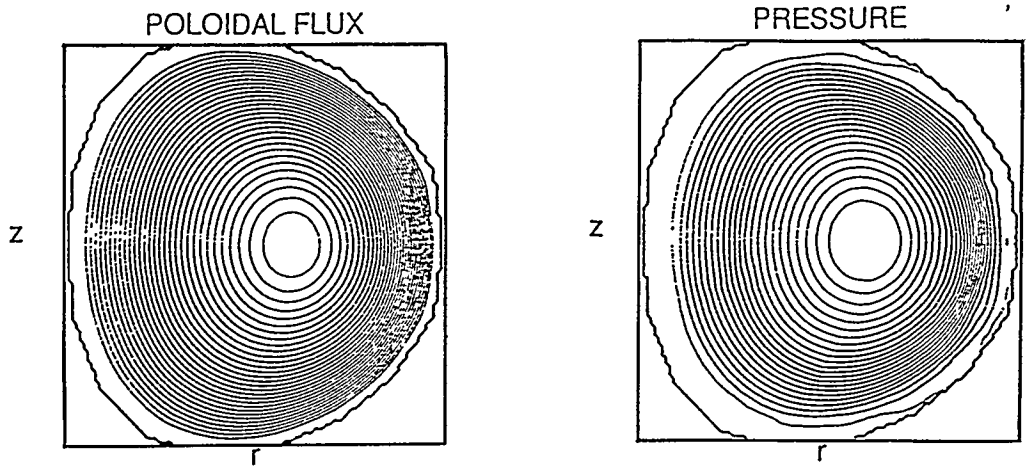

Figure 6-9. Contours of poloidal flux and pressure for the Solov'ev toroidal equilibrium.

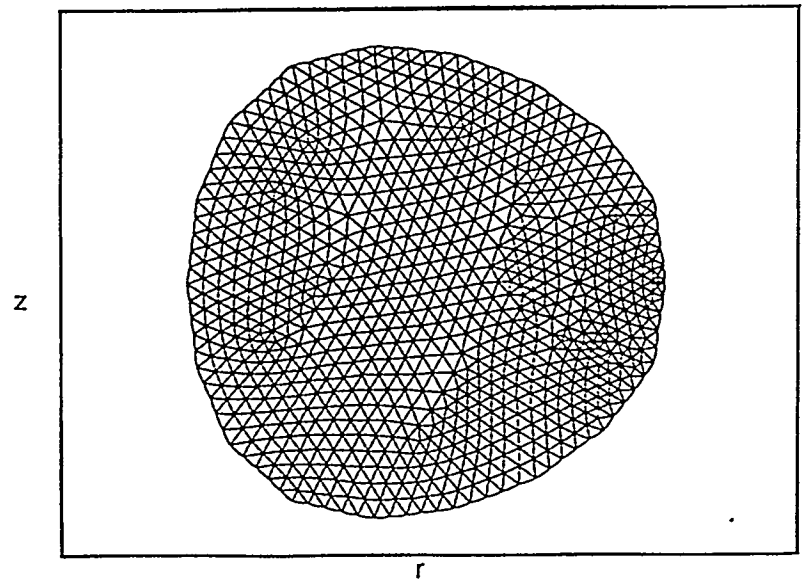

Figure 6-10. Unstructured mesh for the Solov'ev equilibrium.

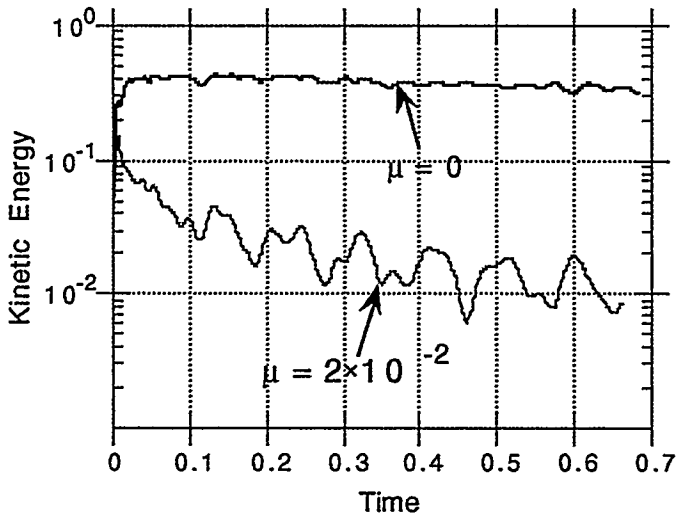

Figure 6-11. Kinetic energy versus time during relaxation of the Solov'ev equilibrium.

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 6-12 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 6-13 for cases with three different values of N_s. This damping rate is summarized in Figure 6-14. We see that the numerical damping rate is approximately linear in $\delta_s \approx (\Delta a_s)^{1/2}$.

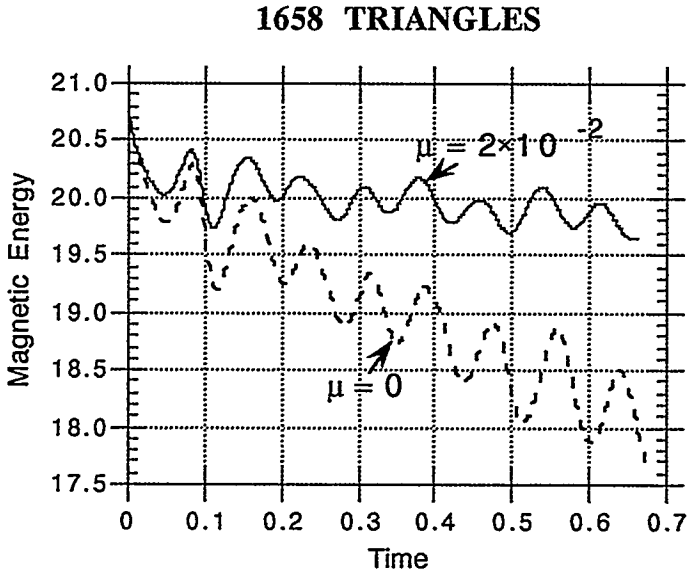

Figure 6-12. Magnetic energy versus time during relaxation of the Solov'ev equilibrium.

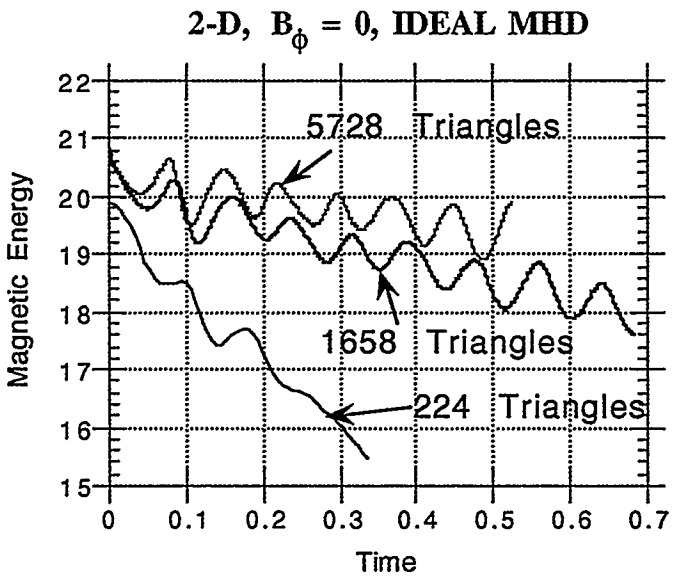

Figure 6-13. Magnetic energy versus time during relaxation of the Solov'ev equilibrium for three different meshes.

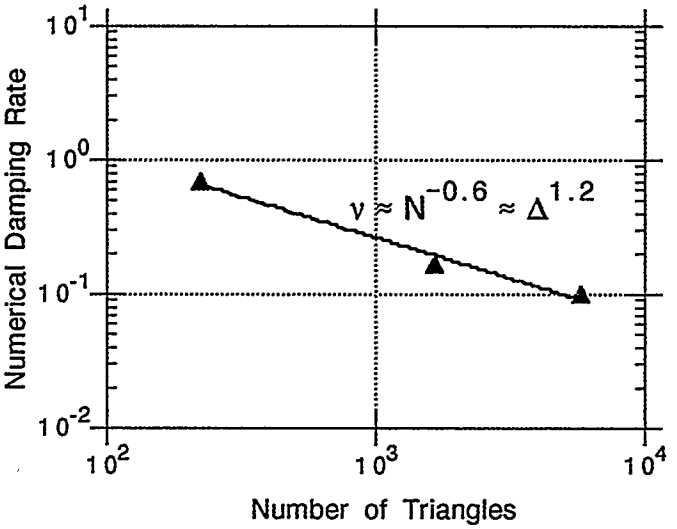

Figure 6-14. Numerical damping rate versus number of triangles.

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 small 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. Even stable equilibria can be driven unstable by diffusive processes[5].

It is to be emphasized that algorithms of the type described in this paper are not the most efficient or accurate way of computing linear stability. Specialized algorithms that find the eigenvalues of the linearized MHD operator are better suited to this problem[21]. Nonetheless, linear stability problems are among the few three-dimensional toroidal problems with known solutions (generally obtained with the specialized algorithms[21]), and are therefore valuable benchmarks for nonlinear, time-dependent algorithms. Physically, linearly unstable MHD modes are of interest only if they impart some observable, and hence finite and nonlinear, perturbation to the physical system. The details of the nonlinear evolution of linearly unstable modes requires that algorithms of the present type be employed.

6.4.1 Solov'ev Equilibrium

Linear Stability and Comparison with Previous Results

The linear stability of the Solov'ev equilibrium described in Section 6.3 to large-scale MHD modes has been extensively studied using specialized eigenvalue techniques[22]. We have developed a "linearized" modification of the TRIM algorithm to study the linear stability of this equilibrium. This modification is possible because of the pseudospectral representation. In these calculations the initial conditions consist of the axisymmetric ($n = 0$) equilibrium, such as Equations (6.4) and (6.5), along with very low amplitude random noise in the velocity field of a single $n = n_0 > 0$ toroidal mode. All other toroidal modes are set to zero initially. The calculation then proceeds as described in Section 5, except that after each time step the amplitudes of all modes with $n \neq n_0$ are reset to zero, and the $n = 0$ (equilibrium) component is restored to its initial value. This effectively disables any nonlinear or quasilinear mode couplings and affords a good approximation to the solution of the linearized equations. The magnetic and kinetic energies of an unstable mode will grow exponentially with time as exp ($2\gamma t$), allowing the growth rate (eigenvalue) γ to be calculated. The self-similar spatial structure of the growing mode defines the eigenvector.

For purposes of comparison it is useful to enumerate the differences between the normalizations of the MHD equations used in TRIM and in the previous work[22]. In Ref. 22 the magnetic field was measured in units of $B_{\phi 0}$, the toroidal field on axis, and time was measured in units of

$$\tau_A = \frac{q_0 R \sqrt{4\pi\rho_0}}{B_{\phi 0}}. \tag{6.6}$$

The quantity q_0 is the safety factor on axis

$$q_0 = \frac{\kappa B_{\phi 0} a^2}{2\Psi_s}, \tag{6.7}$$

where Ψ_s is the (dimensional) value of the poloidal flux at the plasma edge, κ is the elongation (see Section 6.3), and a the minor radius.

In the present formulation (TRIM), distance is measured in units of the major radius R and the magnetic field is normalized so that ψ_s, the (nondimensional) poloidal flux at the plasma edge, is unity. Thus the magnetic field is measured in units of

$$B_0 = \Psi_s / R^2 , \tag{6.8}$$

where R is the major radius. Time is measured in units of

$$\tau_0 = \frac{R\sqrt{4\pi\rho_0}}{B_0} . \tag{6.9}$$

In this normalization, the safety factor q_0 becomes

$$q_0 = \frac{C}{2}\kappa\varepsilon^2 , \tag{6.10}$$

where $\varepsilon = a/R$ is the inverse aspect ratio, and C is the normalized value of the toroidal magnetic field on axis (see Section 6.3).

The normalized growth rates in the two formulations are therefore related by

$$(\gamma\tau_A) = q_0 \frac{B_{\phi 0}}{B_0}(\gamma\tau_0) = \frac{\kappa\varepsilon^2}{2}(\gamma\tau_0) . \tag{6.11}$$

Equation (6.11) can be used to compare the results obtained with TRIM with the those of Ref. 22.

The Solov'ev equilibrium can be completely parameterized in terms of the three nondimensional constants ε (the inverse aspect ratio), κ (the elongation), and q_0 (the safety factor on axis). In Ref. 22, values of $\gamma\tau_A$ were obtained over a range of q_0 for values of $\varepsilon = 1/3$, and $\kappa = 1$ and 2. Here we have primarily focused our attention on the cases with $q_0 = 0.5$, which exhibit robust instability to ideal MHD modes for these values of ε and κ. Special cases with $q_0 = 0$ and $q_0 = 0.8$ will also be described. Also, we have used a boundary condition that corresponds to a perfectly conducting boundary placed at the plasma edge. Thus, only internal (rigid boundary) modes are considered.

The growth rate, or eigenvalue, obtained with any finite representation depends not only on the properties of the axisymmetric equilibrium, but also on numerical parameters that may characterize the algorithm, such as time step, number of spatial zones, artificial viscosity, and resistivity. A precise determination of the growth rate requires that convergence studies be performed to calculate the growth rate in the limit of vanishing time step, viscosity, and resistivity, and increasingly large numbers of spatial zones. Here we have performed such a

convergence study only for the two cases $\varepsilon = 1/3$, $n_0 = 2$, $q_0 = 0.5$, and $\kappa = 1$ and 2. Quantitative comparisons should be made for these cases only. The growth rates obtained for all other parameters should be viewed as imprecise.

The linear stability results for the Solov'ev equilibrium are summarized in Table I. In this table, column 5 contains $\gamma\tau_0$, the normalized growth rate obtained with TRIM, column 6 contains $\gamma\tau_A$, the TRIM growth rate normalized as in Ref. 22 (see Eq. 6.11), and column 7 contains the converged growth rate from Ref. 22. (An asterisk in column 5 indicates that the result is not converged.) The relevant comparison is between columns 6 and 7.

For the converged cases ($q_0 = 0.5$, $n_0 = 2$, $\kappa = 1$ and 2), the growth rates determined from TRIM are about 20% lower than those of Ref. 22, and are consistently lower for the non-converged cases. A quantitative result for the case with $q_0 = 0$, $\kappa = 1$, is not given in Ref. 22, where this mode is identified as "an $m = 0$ mode and not a kink".

The eigenfunctions (spatial structure) for the cases $n_0 = 2$, 3 and 4, $q_0 = 0.5$, and $\kappa = 1$ are shown in Figures 6-15–6-17. In each figure, velocity vectors of the real part of the poloidal velocity (v_r, v_z), and contours of the imaginary part of the toroidal velocity (v_ϕ), for mode n_0 are shown. For the $n_0 = 2$ mode, the poloidal structure is dominantly $m = 1$, while for the $n_0 = 4$ mode, the poloidal structure is dominantly $m = 2$. This is consistent with the value $q_0 = 0.5$, and in agreement with the results of Ref. 22. The eigenfunction for the case $n_0 = 2$, $q_0 = 0$, $\kappa = 1$ is shown in Figure 6-18. It is easy to see the dominant $m = 0$ interchange structure of this mode.

TABLE I.

SOLOV'EV EQUILIBRIUM

Linear Stability Results

R/a	κ	q_0	n_0	$\gamma\tau_0$	$\gamma\tau_A$	$\gamma\tau_A$ (Ref. 22)
3	1	0	2	0.6*	0.03*	(not given)
3	1	0.5	2	2.2	0.122	0.158
3	1	0.5	3	1.0*	0.1*	0.17
3	1	0.5	4	1.56*	0.09*	0.22
3	2	0.5	2	2.0	0.22	0.28
3	2	0.5	3	1.9*	0.21*	0.35
3	2	0.8	1	0.1*	0.011*	0.11

*Result not converged.

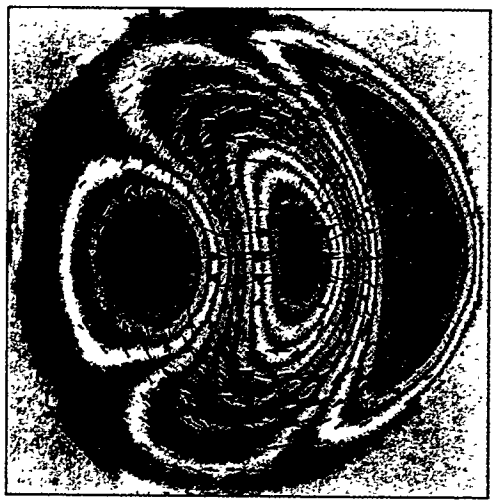

Figure 6-15. Linear eigenfunction for the $n = 2$ mode in the Solov'ev equilibrium with $\kappa = 1$, $q_0 = 0.5$, $R/a = 3$. Velocity vectors display the real part of the poloidal velocity (v_r, v_z), contours display the imaginary part of the toroidal velocity (v_ϕ).

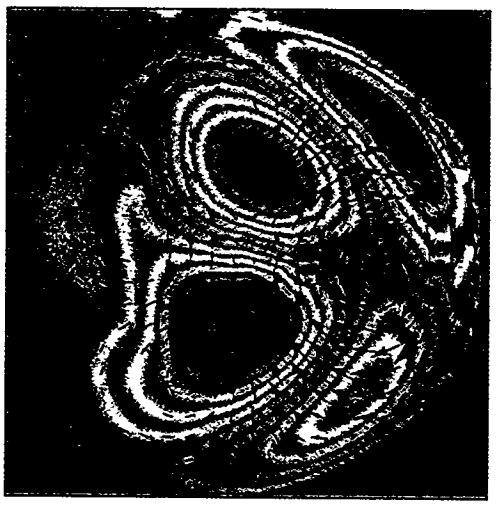

Figure 6-16. Linear eigenfunction for the $n = 3$ mode in the Solov'ev equilibrium with $\kappa = 1$, $q_0 = 0.5$, $R/a = 3$.

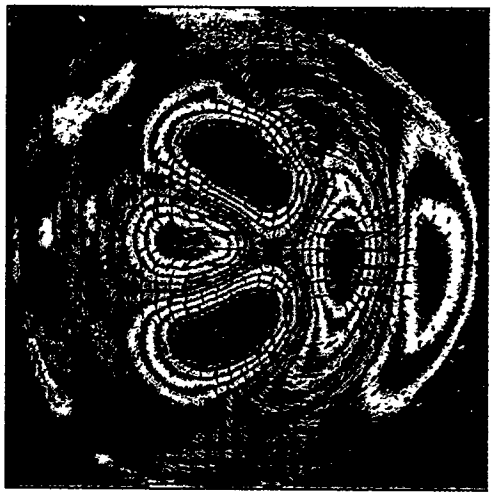

Figure 6-17. Linear eigenfunction for the $n = 4$ mode in the Solov'ev equilibrium with $\kappa = 1$, $q_0 = 0.5$, $R/a = 3$.

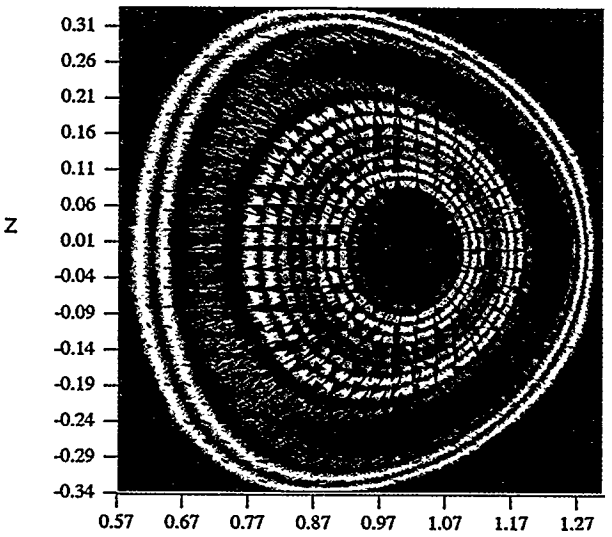

Figure 6-18. Linear eigenfunction for the $n = 2$ mode in the Solov'ev equilibrium with $\kappa = 1$, $q_0 = 0$, $R/a = 3$. Note the $m = 0$ interchange structure.

The equilibrium for the case $\kappa = 2$, $q_0 = 0.5$ is shown in Figure 6-19. Velocity eigenfunctions for the cases $\kappa = 2$, $q_0 = 0.5$, and $n_0 = 2$ and 3 are shown in Figures 6-20 and 6-21. The dominant poloidal mode structure is in agreement with that of Ref. 22. Finally, in Figure 6-22 we display the velocity eigenfunction for the case $\kappa = 2$, $q_0 = 0.8$, $n_0 = 1$. The rigid $m = 1$ displacement of the mode is in contrast with the vortex structure displayed by the other unstable modes found here.

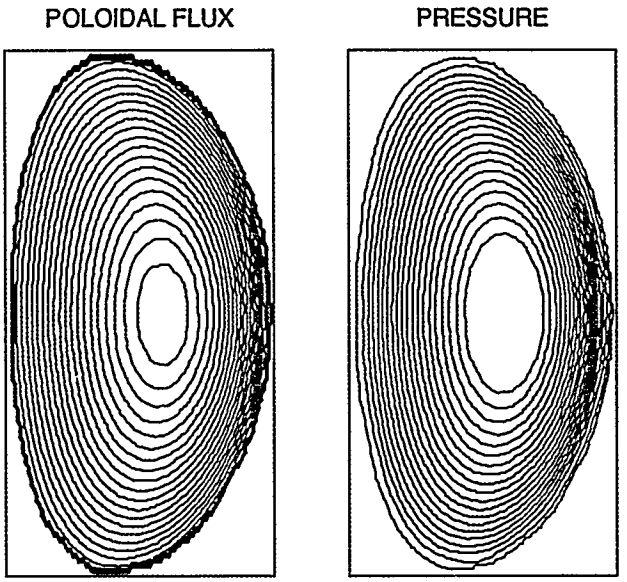

Figure 6-19. Poloidal flux surfaces and pressure contours for the elongated Solov'ev equilibrium $\kappa = 2$, $q_0 = 0.5$, $R/a = 3$.

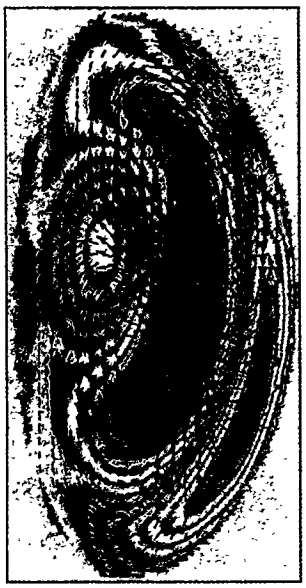

Figure 6-20. Linear eigenfunction for the $n = 2$ mode in the Solov'ev equilibrium with $\kappa = 2$, $q_0 = 0.5$, $R/a = 3$.

Figure 6-21. Linear eigenfunction for the $n = 3$ mode in the Solov'ev equilibrium with $\kappa = 2$, $q_0 = 0.5$, $R/a = 3$.

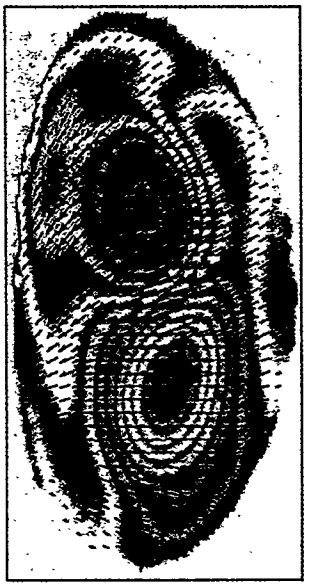

Figure 6-22. Linear eigenfunction for the $n = 1$ mode in the Solov'ev equilibrium with $\kappa = 2$, $q_0 = 0.8$, $R/a = 3$. Note the rigid $m = 1$ displacement of this mode.

Effect of Time Step on Linear Growth Rate

As discussed in Section 5, TRIM uses the semi-implicit method[8] to achieve time steps in excess of that set by numerical stability constraints. As shown in Ref. 11, the semi-implicit method reduces the characteristic frequency of a mode with wave number k by a factor

$$\frac{\omega}{\omega_0} = \frac{1}{\sqrt{1+\alpha k^2 \Delta t}}, \tag{6.12}$$

where α is the semi-implicit coefficient and ω_0 is the frequency obtained by an explicit calculation with $\Delta t = \Delta t_{CFL}$. Using Equation 5.17, this expression can be rewritten as

$$\frac{\omega}{\omega_0} = \frac{1}{\sqrt{1+\sigma\left[\left(\frac{\Delta t}{\Delta t_{CFL}}\right)^2 - 1\right]\left(\frac{k}{k_{max}}\right)^2}}, \tag{6.13}$$

where σ is a constant of order unity and k_{max} is the maximum value of k that can be resolved on the grid. For unstable modes, the frequency ω becomes the growth rate γ.

For the case $n_0 = 2$, $\kappa = 1$, and $q_0 = 0.5$ (see Fig. 6-16), we have $k_{max} = 164$ (corresponding to 2090 triangles), $k \approx 15$ (corresponding to $\lambda_r \approx \lambda_z \approx 2a \approx 0.6$), and $\sigma = 1.5$. This case was run with a small amount of artificial viscosity, $\nu = 10^{-4}$. In Figure 6-23 we plot the ratio γ/γ_0 obtained from both TRIM and from Equation 6.13 as a function of $\Delta t/\Delta t_{CFL}$ for this case. These results substantially confirm the effect of the semi-implicit method on the growth rate. The discrepancy between the two curves may be due to the artificial viscosity or other numerical damping inherent in the algorithm.

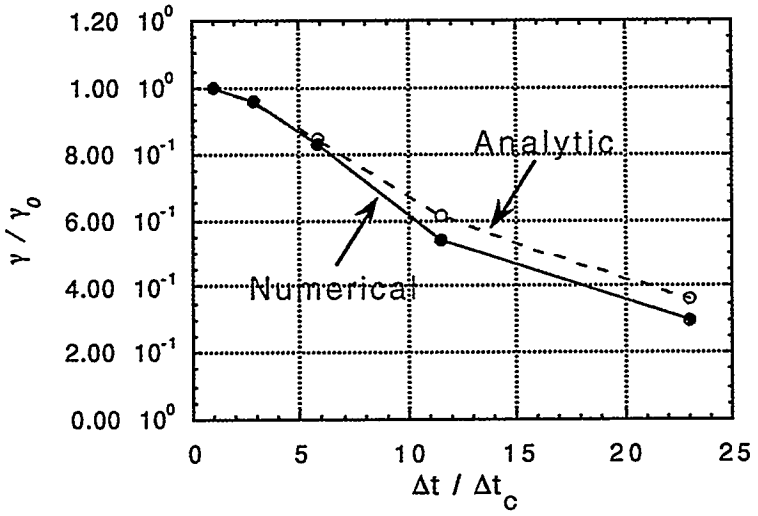

Figure 6-23. The effect of the semi-implicit method on the linear growth rate obtained with TRIM for the case $\kappa = 1$, $q_0 = 0.5$, $R/a = 3$, $n = 2$, as a function of the time step (in excess of the CFL limit).

Nonlinear Results

The linear results described above were obtained by removing all nonlinear interactions, freezing the $n = 0$ component of the solution, and allowing only a single mode with $n = n_0$ to evolve. To obtain the full nonlinear dynamical evolution of the unstable equilibrium, all modes with $n > 0$ are initially perturbed, all nonlinear interactions are restored, and the $n = 0$ component is allowed to evolve under the influence of unbalanced forces, resistivity, viscosity, and nonlinear effects. Since these cases may involve large amplitude displacements and considerable dynamics, larger values of resistivity and artificial viscosity are used than in the strictly linear results. Typical values are $S = 10^4$ and $\nu = 10^{-2}$. Total toroidal flux is conserved.

In Figure 6-24 we plot the time evolution of the kinetic energy in the modes $1 \le n \le 5$ (corresponding to 16 toroidal mesh points) for the case $\kappa = 1$, $q_0 = 0.5$, $R/a = 3$. The time step is such that $\Delta t/\Delta t_{CFL} = 11.5$. Because of the finite resistivity, the total toroidal current decays and its profile peaks during the evolution, thus altering the linear stability properties of the discharge. The $n = 2$ mode is linearly unstable, grows to finite amplitude, and saturates. The $n = 4$ mode, which was found to be linearly unstable by both TRIM and in Ref. 22, exhibits initial exponential growth at approximately twice the rate of the $n = 2$ mode. This, and the delayed onset of the mode, implies that this mode is driven nonlinearly by the $n = 2$ mode rather than by inherent linear instability. The change in the linear stability properties of this mode may be due to the modification of the $n = 0$ component by resistive diffusion, but we have not verified this conjecture. The $n = 1$ and $n = 5$ modes exhibit complete linear stability. The small amplitude increases at late time in these modes is due to nonlinear spectral broadening from the saturation of the $n = 2$ mode. The $n = 3$ mode shows some small indication of linear instability late

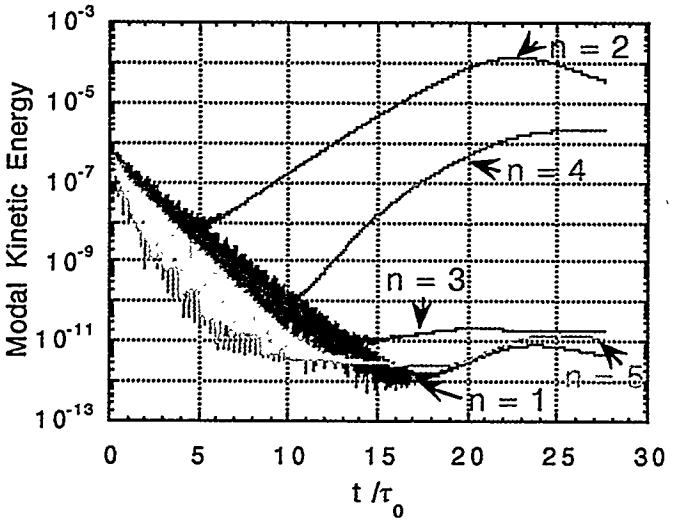

Figure 6-24. Kinetic energy in the modes $1 \le n \le 5$ versus time for the nonlinear evolution of the Solov'ev equilibrium with $\kappa = 1$, $q_0 = 0.5$, $R/a = 3$.

in the calculation but prior to $n = 2$ saturation. We cannot rule out nonlinear spectral broadening as the cause of this increase. In Figure 6-25 we plot contours of the pressure in the saturated state at four different toroidal locations spanning one-half of the torus. The predominantly $n = 2$, $m = 1$ helical displacement of the plasma column is evident.

The same calculation has been performed for the case $\kappa = 2$, $R/a = 3$, $q_0 = 0.5$, with 32 toroidal mesh points (corresponding to toroidal mode numbers $0 \le n \le 10$ after dealiasing). The magnetic energy in the modes $1 \le n \le 10$ is shown in Figure 6-26. The modes $n = 2$ and $n = 3$ exhibit robust linear instability. The $n = 4$ mode also indicates initial linear instability, but makes a transition to being nonlinearly driven by the $n = 2$ mode later in the calculation. The $n = 1$ and $n = 5$ modes are driven by the nonlinear interaction of the $n = 2$ and $n = 3$ modes, and the $n = 6$ mode is nonlinearly driven by the $n = 3$ mode. All other modes appear to be driven by nonlinear spectral broadening. The finite amplitude of all modes at the end of the calculation probably indicates that more toroidal mesh points (modes) are required for proper resolution of the nonlinear state. In Figure 6-27 we plot contours of the pressure in the saturated state at four different toroidal locations spanning one-half of the torus. Again, the helical displacement is topologically $m = 1$, $n = 2$ dominant.

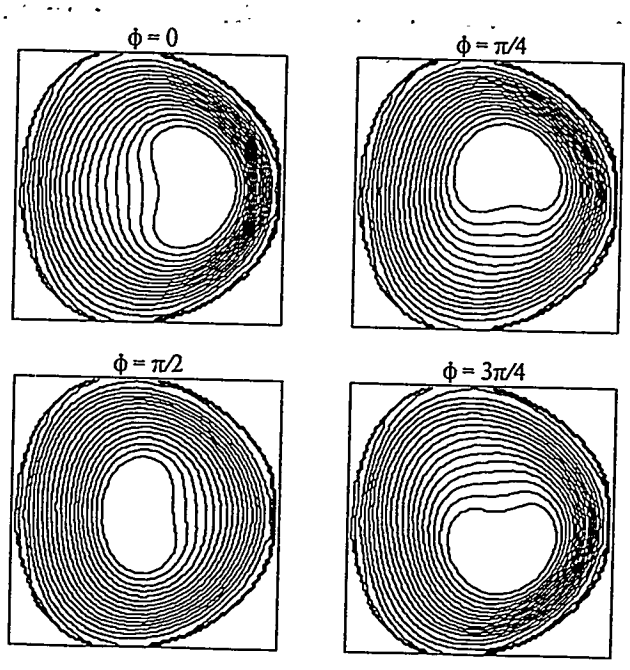

Figure 6-25. Pressure contours in the poloidal plane at four different toroidal locations in the nonlinearly saturated state of the Solov'ev equilibrium with $\kappa = 1$, $q_0 = 0.5$, $R/a = 3$.

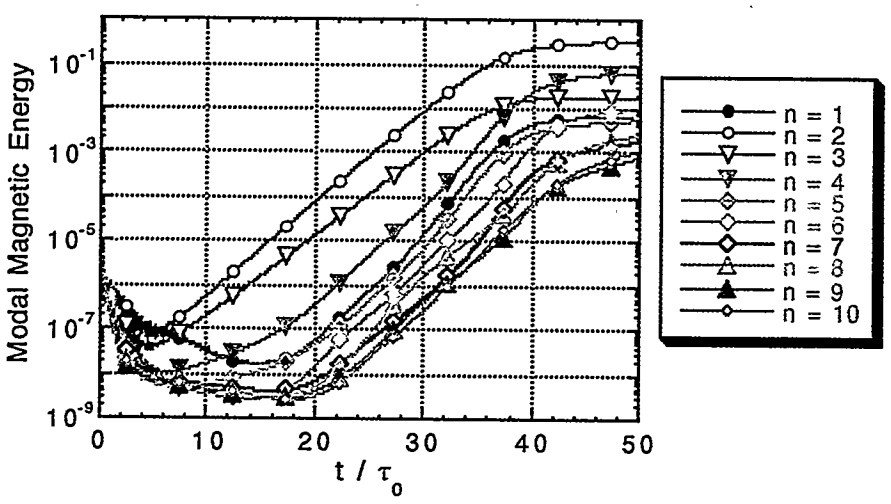

Figure 6-26. Magnetic energy in the modes $1 \leq n \leq 10$ versus time for the nonlinear evolution of the Solov'ev equilibrium with $\kappa = 2$, $q_0 = 0.5$, $R/a = 3$.

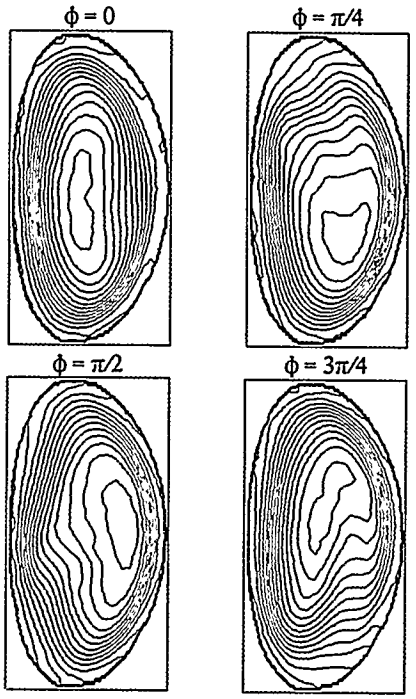

Figure 6-27. Pressure contours in the poloidal plane at four different toroidal locations in the nonlinearly saturated state of the Solov'ev equilibrium with $\kappa = 2$, $q_0 = 0.5$, $R/a = 3$.

6.4.2 ITER Equilibrium

In Figure 6-28a,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[23]. The outer boundary is the separatrix, or last closed flux surface; it is intended that the plasma is confined within this surface. This equilibrium has been constructed to be unstable to an internal kink mode with toroidal mode number $n = 2$. The safety factor profile is shown in Figure 6-29. The linear instability has been computed with the GATO code[21], which directly solves the resulting linear eigenvalue problem.

The unstructured mesh inside the separatrix with $N_s = 5728$ is shown in Figure 6-30. The equilibrium is initialized to this mesh by cubic spline interpolation, and the resulting force imbalance is resolved with viscous damping as described in the Appendix. 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.

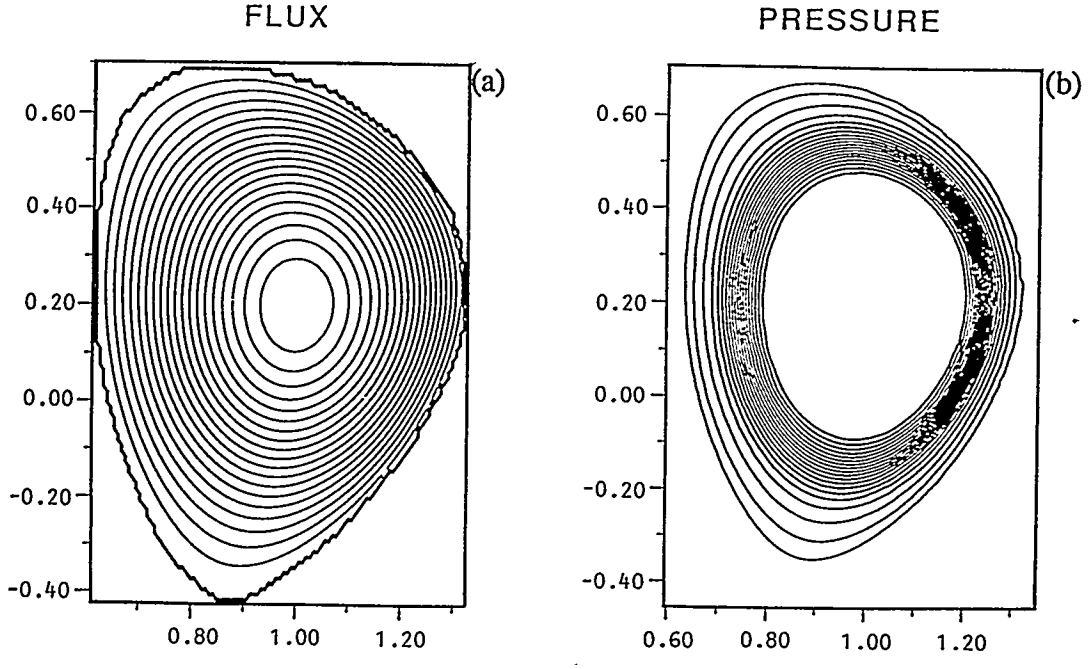

Figure 6-28(a,b). Contours of poloidal flux and pressure for an ITER equilibrium.

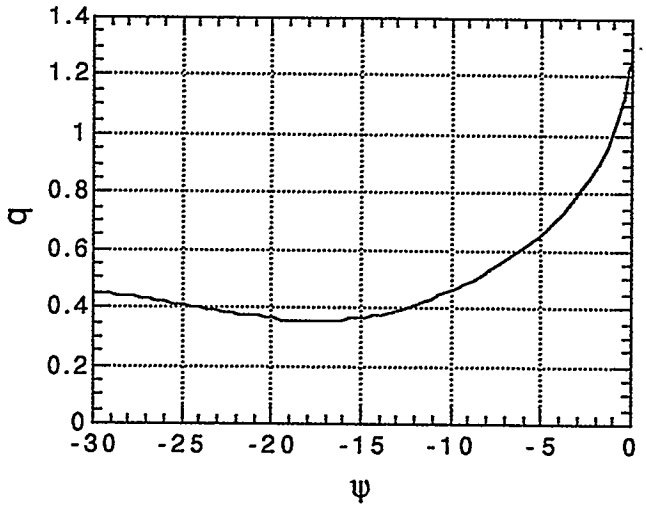

Figure 6-29. The safety factor q (ψ) for the ITER equilibrium.

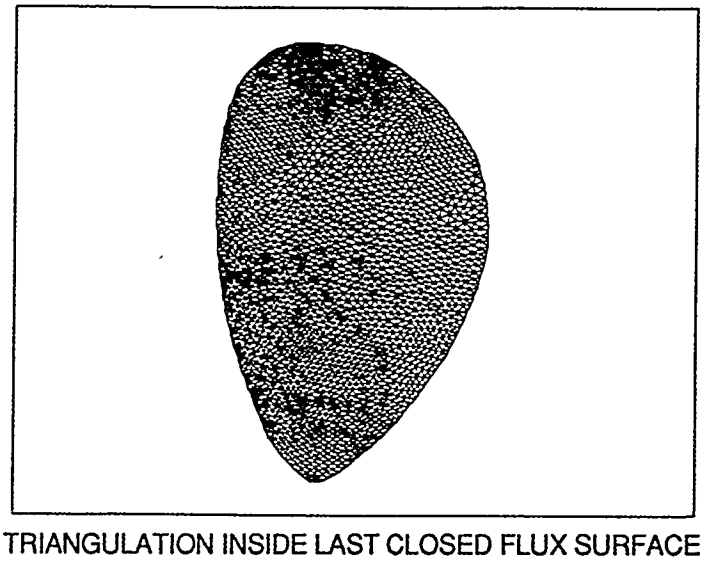

Figure 6-30. Unstructured mesh for the ITER equilibrium.

The kinetic energy in the $n = 1$ and $n = 2$ modes are shown in Figure 6-31. The $n = 2$ mode is unstable, and the $n = 1$ mode is stable, in agreement with linear calculations[23].

The linear eigenmode for the poloidal velocity is shown in Figure 6-32, where the poloidal velocity vectors are shown 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 twice around the torus, as required by an $n = 2$ mode.

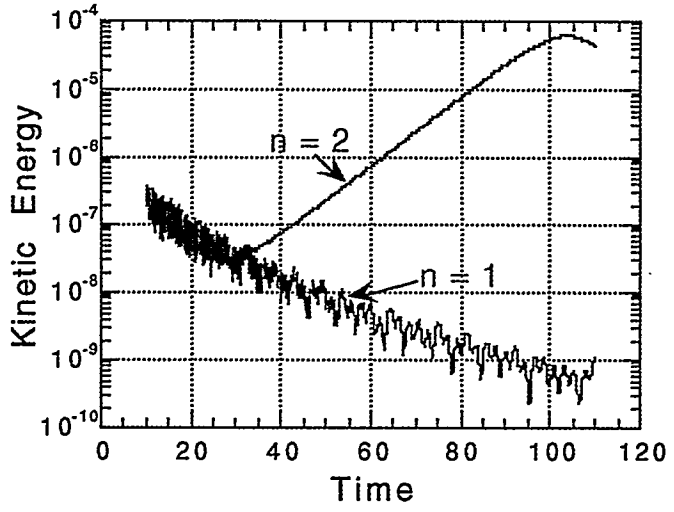

Figure 6-31. Kinetic energy versus time for the $n = 1$ and $n = 2$ modes in the ITER equilibrium.

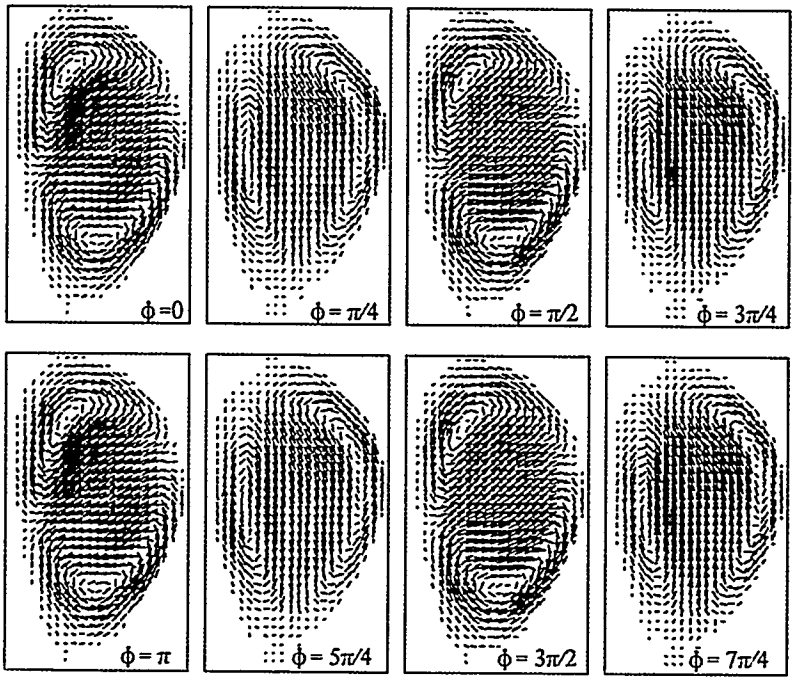

Figure 6-32. Poloidal velocity in the linear eigenmode at 8 different toroidal locations.

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.

6.4.3 Resistive Instability in a Torus

The results presented in Sections 6.4.1 and 6.4.2 described ideal instabilities, i.e., unstable normal modes of the ideal (infinitely conducting) MHD equations. However, some of the most important instabilities in toroidal fusion devices exist only in the presence of finite resistivity. As a result of these modes the magnetic field configuration can change its topological properties, which are otherwise invariant in ideal MHD. These resistive instabilities[24] have no counterpart in ideal MHD, and grow on a time scale that is a hybrid of the Alfvén and resistive time scales. The computation of these modes has been a primary focus of fusion computations for over 20 years.

We have begun to apply TRIM to resistive instabilities in a torus. We have studied the linear stability of a toroidal equilibrium whose stability properties are well known[25]. For this case, the poloidal cross section of the plasma is circular, and the q-profile is given by

$$q(r) = 2\left(\frac{1+(r/r_0)^8}{1+(r_{s2}/r_0)^8}\right)^{1/4} \qquad (6\text{-}14)$$

where r is the minor radius (measured from the center of the circular poloidal cross section), r_{s2} is the radius of the $q = 2$ surface, and r_0 is the width of the current channel. We use $r_{s2} = 0.7$ and $r_0 = 0.6$, which corresponds to Run 1, Table I of Ref. 25. The particular equilibrium has been supplied in numerical form[26]. In Ref. 25, the Lundquist number at the $q = 2$ surface was $S = 2 \times 10^4$. For our initial calculations, we have used an enhanced resistivity that gives $S = 10^3$ at the $q = 2$ surface.

We perturb the initial equilibrium with random noise, and consider a linear case with toroidal mode number $n = 1$. We find an exponentially growing instability with a growth rate of $\gamma\tau_{Ap} = 0.01$ (where τ_{Ap} is the poloidal Alfvén time). This is to be compared with the result $\gamma\tau_{Ap} = 0.017$ of Ref. 25 with $S = 2 \times 10^4$. The effect of this resistive instability on the magnetic field topology is shown in Figure 6-33, where we the successive intersections of four different magnetic field lines with the poloidal plane (a Poincaré plot). The field line integration was performed with the code TUBE[27]. The magnetic islands that characterize resistive instabilities correspond topologically to a poloidal mode number $m = 2$, in agreement with Ref. 25.

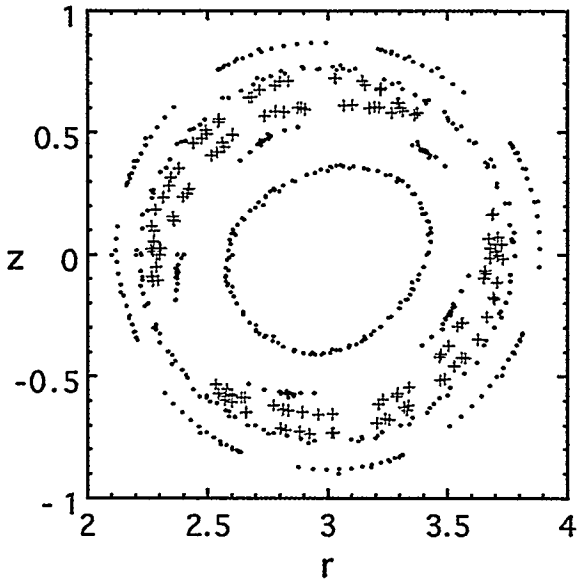

Figure 6-33. Intersection of four different field lines with the poloidal plane showing the magnetic island that results from an unstable $n = 1$ resistive instability. The topologically $m = 2/n = 1$ structure is evident.

7.0 SUMMARY AND DISCUSSION

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. This allows axisymmetric configurations with arbitrarily complex poloidal geometry to be accurately represented. Boundaries in the poloidal plane need not correspond to flux surfaces or be constrained by coordinate systems. 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 verified. 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 to both the hydrodynamic and MHD cases. Toroidal force balance has been computed by viscous damping of Alfvén and sound waves. Linear growth and nonlinear saturation of three-dimensional kink modes in two analytic equilibria, and in a highly elongated toroidal equilibrium representative of the ITER design, has been computed. Converged linear growth rates agree with previous linear stability calculations to within twenty percent. The nonlinear evolution of these modes has shown nonlinear mode coupling and

spectral broadening, and has demonstrated the utility of the semi-implicit method of time integration for these calculations. The linear growth of a resistive tearing instability in toroidal geometry has also been calculated.

We view the work presented here as a first step toward the development of a flexible, accurate algorithm for time-dependent MHD in arbitrary geometry. Several issues have arisen in the course of our investigation that we believe require further research. These are enumerated below.

The first issue concerns the most efficient and accurate use of the primary and dual meshes, and the placement of dependent variables on them. In this work we have used the Delaunay triangles as the primary control volume, and have chosen to define all volume densities (momentum, mass, and energy) at their centroids. All physical boundaries consist of triangle edges. We have made not made use of the dual control volume elements consisting of the polygons centered at triangle vertices with edges connecting triangle centroids. (The edges of these polygons are used to define components of the magnetic field and current density.) With the present scheme differential operators such as the gradient and the Laplacian acting on triangle-centered densities couple more than nearest neighbor triangles. Our experience with rectangular, structured meshes[11] indicates that the use of staggered, overlapping volume elements leads to the most compact, accurate, and physically motivated algorithms. In those methods, the pressure and momentum are not collocated as they are here, but are defined at the centers of the staggered primary and dual meshes, respectively. We have not experienced any severe problems that can be directly attributed to the non-compact formulation described in this paper, but the present algorithm seems to require somewhat more artificial viscosity to assure robustly stable computations than methods that use rectangular, structured meshes. Our initial attempts to formulate an exactly equivalent method using triangles and polygons has led to problems in consistently defining boundary edges and applying boundary conditions.

The second issue concerns interpolation. In the present algorithm, interpolation from centroids to vertices, vertices to centroids, centroids to edges, edges to vertices, and vertices to edges are all required. The form of the interpolation can affect the accuracy and stability of the calculation. As described in Section 4.3, some of these interpolation schemes can become quite complicated and can lead to coupling beyond nearest neighbors. We have not devised a method for MHD using either rectangular or triangular meshes that does not require some interpolation (or averaging) from one grid to another. Interpolation is required during mesh refinement and coarsening. The number of interpolations per time step is also affected by the choice of primary and dual meshes. The issues of accuracy and required number of interpolations, and their affect on the performance of the algorithm, must be better understood.

The third issue concerns the criteria used for dynamic refinement and coarsening of the mesh. In Section 2.7 we presented a criterion for adding or deleting triangles that was based on the normalized average Laplacian operator. For the results presented in Section 6, we have used this criterion in conjunction with the mass density. Other criteria, such as the normalized gradient of a variable or the divergence of a flux, have been used in computational hydrodynamics[9]. Other

problems may require still other criteria. For example, criteria based on current density may be useful for magnetic reconnection problems. The proper criteria for fusion MHD is yet to be determined.

The fourth issue concerns the structure of the mesh itself. As described in Section 2.4, Delaunay triangles have many desirable properties. The algorithm is most accurate for equilateral triangles, for then the centroid dual mesh and the Voronoi dual mesh are equivalent. Deviations from equiangularity introduce errors, and highly obtuse triangles can lead to spikes in high order derivatives, such as current density. Mesh refinement can lead to large variations in triangle size and shape over the mesh, even though the Delaunay triangulation maximizes equiangularity in a global sense. These large variations can in turn affect the diagonal dominance of the Laplacian operator and cause the conjugate gradient algorithm to fail to converge. What is needed is a method for systematically redistributing the vertices in the poloidal plane to assure that all triangles are at least acute, and that variations in mesh size are smoothly distributed in space. Clearly more fundamental work needs to be done in this area.

Finally, we have used low order approximations for the calculation of interface fluxes. The accurate computation of these fluxes has occupied the attention of computational hydrodynamicists for several years, and it has proven to be a crucial issue in the accurate engineering application of these methods. Improvements in these methods for MHD must eventually be addressed. However, in light of the other fundamental issues discussed in this section, we do not feel that this is a priority for fusion applications.

ACKNOWLEDGEMENT

Many scientists have contributed to the work described in this paper. The Principal Investigator was Dr. D. D. Schnack, who devised the MHD algorithm and was responsible for the overall project, including any shortcomings it may contain. The application of unstructured meshes and dynamic mesh adaption to fusion MHD was suggested by Dr. A. Drobot. The logical data structure for the unstructured triangular mesh, and the routines for dynamic mesh adaption, were developed and written by Dr. I. Lottati and Dr. S. Eidelman, who kindly supplied them from their hydrodynamics code. This work would have been impossible to even begin, much yet complete, without this contribution. The interface between the basic MHD algorithm and the dynamic adaption algorithm was designed, written, and debugged by Dr. P. Satyanarayana, who also carried out many of the benchmark calculations. The conjugate gradient algorithm used for matrix inversion on the unstructured mesh was written by Dr. Z. Mikić, who also contributed to the development of the basic MHD algorithm. Valuable discussions on many aspects of unstructured meshes and mesh adaption were held with our colleagues Dr. H. R. Strauss, Dr. A. Koniges, Dr. A. Glasser, Dr. A. Kuprat, Dr. D. Longcope, and Dr. D. C. Barnes. Dr. A. Turnbull provided the ITER equilibrium and linear stability results, and Dr. A. Aydemir provided the equilibrium used for the calculation of resistive instabilities. Finally, none of these results would be communicated to anyone without the thorough and professional efforts of Ms. A. Miklovic.

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. I. Lottati and S. Eidelman, *App. Num. Math* 14, 353 (1994).
10. D. Nielson, private communication.
11. D. D. Schnack *et al., J. Comp. Phys.* 70, 330 (1987).
12. T. J. Barth, "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations", in Ref. 6, above.
13. P. J. Green and R. Sibson, *The Computer Journal* 21, 191 (1983).
14. R. Löhner, K. Morgan, and O. C. Zienkiewicz, *Comp. Math. Appl. Mech. Eng.* 51, 441 (1985).
15. R. Löhner, "The Efficient Simulation of Strongly Unsteady Flows by the Finite Element Method," AIAA-87-055 (1987).
16. 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.
17. Z. Mikić and E. C. Morse, *J. Comp. Phys.* 61, 154 (1985).
18. G. Sod, *J. Comp. Phys.* 27, 1 (1978).
19. M. Brio and C. C. Wu, *J. Comp. Phys.* 75, 400 (1988).
20. L. S. Solov'ev, *Sov. Physics JETP* 26, 400 (1968).
21. R. Gruber *et al., Comput. Phys. Commun.* 24, 363 (1981).
22. D. Berger, L. C. Bernard, R. Gruber, and F. Troyon, *J. App. Math. and Physics (ZAMP)* 31, 113 (1980).

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

24. H. P. Furth, J. Killeen, and M. N. Rosenbluth, *Phys. Fluids* **16**, 1054 (1973).

25. B. V. Waddell, B. Carreras, H. R. Hicks, and J. A. Holmes, *Phys. Fluids* **22**, 896 (1979)

26. A. Y. Aydemir, private communication (1995).

27. A. A. Mirin, D. R. Martin, and N. J. O'Neill, *Comput. Phys. Commun.* **54**, 183 (1989).