An Adaptive Finite Element Method for Magnetohydrodynamics

H. R. Strauss and D. W. Longcope
Courant Institute of Mathematical Sciences
New York University

August 11, 1998

Abstract

A finite element discretization for two dimensional MHD is described. The elements are triangles with piecewise linear basis functions. The main computational difficulty is the accurate calculation of the current. The most effective solution is to employ a current - vorticity advection formulation of the equations. Acceptable results can also be obtained with a two - step calculation of the current from the vector potential. Mesh operations are described to reconnect and refine the mesh adaptively in the vicinity of nearly singular currents, to improve magnetic flux conservation. Example computations of the coalescence instability, tilt mode and divertor tokamak equilibrium, validating and illustrating the method, are presented. The simulations show the formation of current sheets, with the current density increasing exponentially in time. During this increase, the grid of initially $\sim 10^4$ points adapts to provide resolution comparable to a uniform grid of up to 1.6×10^8 grid points.

1 Introduction

Finite element, unstructured mesh methods are now just beginning to be used in magnetohydrodynamics (MHD) computations. Unstructured mesh methods have become a leading approach in computational fluid dynamics for two main reasons.

First, they allow adaptive local mesh refinement. MHD motion tends to develop sharp structures: nearly discontinuous magnetic field and sharply localized, intense current sheets. We present simulations showing the formation of current sheets, with the current density increasing exponentially in time. If the resolution is inadequate, truncation error can cause artificial numerical dissipation and magnetic reconnection. Therefore it is important to refine the grid as much as possible where current sheets form. This should be done adaptively, because the location of the sheets can change, as in the case of the tilt mode.

A second feature of unstructured mesh methods is their ability to fit irregular boundaries. Applications to modelling laboratory plasmas may require the use of irregularly shaped boundaries. Although this is an important feature, in this paper we emphasize adaptive mesh refinement to resolve current sheets.

In the following, we first list the incompressible, two dimensional MHD equations. We then give the standard stream function - vorticity advection form of the equations, as well as the symmetrized current - vorticity advection formulation. The equations can be discretized using piecewise linear, triangular finite elements. Three sparse matrices, the mass matrix, stiffness matrix, and bracket tensor, arise in the discretization. Their construction and assembly is discussed. The stiffness matrix can cause a convergence problem in computing the current, for which we give two possible cures.

The most effective cure is to use symmetrized MHD equations, in which vorticity and current are time advanced, and the potentials are found by solving Poisson equations. The other approach is to use a modified stiffness matrix with a wider stencil, having acceptable convergence properties.

Adaptive gridding is done with two mesh operations: splitting pairs of triangles into four triangles; and the inverse operation of combining four triangles into two.

After describing a new finite element code, FEMHD, based on these algorithms, we give examples of its use. We verify that the code reproduces previously known solutions for the coalescence and tilt instability. We carry out simulations of current sheet formation caused by these instabilities. We note that similar results on current sheet formation have recently been obtained using adaptive structured mesh methods [10]. This approach also used symmetrized MHD equations in the Elsässer form, to be given below.

2 MHD

Magnetohydrodynamics (MHD) is the fluid dynamics of conducting fluid or plasma, coupled with Maxwell's equations. The fluid motion induces currents, which produce Lorentz body forces on the fluid. Ampere's Law relates the currents to the magnetic field. The MHD approximation is that the electric field vanishes in the moving fluid frame, except for possible resistive effects.

MHD is described by higher order system of partial differential equations than fluid dynamics. It admits additional waves, the Alfvén waves and their instabilities. A typical feature of MHD is the tendency to form a singular current density. Current sheets, in the presence of resistive dissipation, are associated with the breaking and reconnecting of magnetic field lines.

Various approaches have been used in computational MHD to reduce numerical magnetic dissipation. Lagrangian and partially Lagrangian methods [3] are less diffusive than Eulerian methods, but require substantial rezoning for sheared or straining flow. Mixed finite difference and spectral discretizations have been very effective in dealing with reconnection in periodic geometry [4]. Finite difference codes with nonuniform, Cartesian product grids have also been successful in reconnection simulations in which intense current sheets are aligned with the grid [1]. These approaches, while effective in specialized geometric configurations, are not able to adaptively add grid refinement when current sheets form at arbitrary locations on the mesh. Recently, adaptive structured mesh methods have been successfully applied to these problems [10]. However, these methods are not particularly effective with arbitrarily shaped boundaries. For these reasons we are led to try adaptive unstructured mesh methods. Unstructured meshes have also been used in other computational MHD approaches [11]. We have chosen to apply these methods to two dimensional, incompressible MHD, using a stream function approach to enforce the divergence free conditions on the magnetic and velocity fields. Our approach differs in the use of a symmetrized form of the equations to eliminate difficulties with the calculation of the current density.

The incompressible MHD equations are:

$$\frac{\partial}{\partial t}\mathbf{B} = \nabla \times (\mathbf{v} \times \mathbf{B}) \tag{1}$$

$$\rho\frac{\partial}{\partial t}\mathbf{v} = -\rho\mathbf{v}\cdot\nabla\mathbf{v} + (\nabla \times \mathbf{B}) \times \mathbf{B} + \rho\mu\nabla^2\mathbf{v} \tag{2}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{3}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{4}$$

where $\mathbf{B}$ is the magnetic field, $\mathbf{v}$ is the velocity, $\rho = \rho_0$ is the density, assumed constant, and μ is the viscosity. The equations can be made dimensionless, by normalizing B to a reference magnetic field strength B_0, ρ to ρ_0, v to the Alfvén velocity $v_A = B_0/\sqrt{\rho_0}$, length to L, and time to the Alfvén time, L/v_A. To enforce incompressibility, it is common to introduce stream functions:

$$\mathbf{v} = \left(\frac{\partial \phi}{\partial y}, \ -\frac{\partial \phi}{\partial x} \right) \tag{5}$$

$$\mathbf{B} = \left(\frac{\partial \psi}{\partial y}, \ -\frac{\partial \psi}{\partial x} \right) \tag{6}$$

In two dimensions, with incompressible flow, the MHD equations can be written

$$\frac{\partial}{\partial t}\Omega + [\ \Omega,\ \phi\ ] = [\ C,\ \psi\ ] + \mu \nabla^2 \Omega \ , \tag{7}$$

$$\frac{\partial}{\partial t}\psi + [\ \psi,\ \phi\ ] = 0. \tag{8}$$

$$\nabla^2 \phi = \Omega \tag{9}$$

$$C = \nabla^2 \psi \tag{10}$$

where the two dimensional Laplacian is

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}.$$

and

$$[\ a,\ b\ ] = \frac{\partial a}{\partial x}\frac{\partial b}{\partial y} - \frac{\partial a}{\partial y}\frac{\partial b}{\partial x}$$

The left hand side of (7), along with (9) is the familiar vorticity - stream function formulation of two dimensional incompressible hydrodynamics. The right hand side of (7) comes from the Lorentz force with current density C,, and the viscosity with coefficient μ. Eq. (8) is from Ohm's Law and Faraday's Law and represents the conservation of magnetic flux ψ. The z component of the magnetic field does not enter (7) - (10), so its evolution is not followed. A large, nearly constant B_z is often invoked to justify the incompressible approximation.

The MHD equations conserve energy and magnetic flux. Since the magnetic flux function is advected with the flow, any function of ψ is a constant of the motion. The energy, $\mathcal{E}$, can be shown to be conserved by pre multiplying the Ω evolution equation by ϕ, and the ψ evolution equation by C, and integrating by parts, or working directly from the primitive form of the equations. One obtains

$$\frac{\partial}{\partial t}\mathcal{E} = -\mu \int d^2x\Omega^2.$$

where

$$\mathcal{E} = \frac{1}{2}\int d^2x \left(\mathbf{v}^2 + \mathbf{B}^2\right). \tag{11}$$

assuming either Dirichlet boundary conditions, with ϕ, ψ constant on the boundary, Neumann conditions with the normal derivatives of ϕ, ψ equal to zero, or periodic boundary conditions.

3 Symmetrization of Equations

The evolution of the magnetic and velocity fields are treated in a non symmetric way in the standard formulation above. The velocity is advanced through the vorticity, while the magnetic field is advanced via the magnetic potential. This can cause numerical problems when the equations are solved on an irregular mesh, such as the unstructured, adaptive grids we will use later. It is desirable to formulate the equations in a more symmetrical manner, in which the current and vorticity are time advanced.

Instead of solving eqs. (8) and (10), we take the Laplacian of (8) and use (9), (5), and (6). This yields an equation for the current, analogous to (7) for the vorticity,

$$\frac{\partial}{\partial t}C = [\ \phi,\ C\ ] + [\ \Omega,\ \psi\ ] + 2[\frac{\partial \phi}{\partial x}, \frac{\partial \psi}{\partial x}] + 2[\frac{\partial \phi}{\partial y}, \frac{\partial \psi}{\partial y}] \tag{12}$$

$$\nabla^2 \psi = C \tag{13}$$

Eqs. (12), (13) are solved along with eqs. (7) and (9). The equations are now symmetrical, in the sense that the source functions Ω and C are time advanced, and the potentials ϕ and ψ are obtained at each time step by solving Poisson equations (9) and (13). Dirichlet, Neumann, or periodic boundary equations are applied to the potentials.

The equations can be given an even more symmetric form using the Elsässer variables $\mathbf{v} \pm \mathbf{B}$ for which:

$$\phi^{\pm} = \phi \pm \psi$$

$$\Omega^{\pm} = \Omega \pm C.$$

Combining the evolution equations for Ω and C gives (dropping the viscous term)

$$\frac{\partial \Omega^{\pm}}{\partial t} = [\phi^{\mp}, \Omega^{\pm}] \pm [\frac{\partial \phi^{\mp}}{\partial x}, \frac{\partial \phi^{\pm}}{\partial x}] \pm [\frac{\partial \phi^{\mp}}{\partial y}, \frac{\partial \phi^{\pm}}{\partial y}] \tag{14}$$

$$\nabla_{\perp}^2 \phi^{\pm} = \Omega^{\pm}. \tag{15}$$

4 Finite Element method

The mesh points of the grid are the vertices of triangles, located at points $\mathbf{r}_i$. We use a finite element discretization, introducing basis functions $\lambda_i(\mathbf{r})$, which in the present work are piecewise linear over each triangle, and satisfy $\lambda_i(\mathbf{r}_j) = \delta_{ij}$. The basis function λ_i is equal to unity on vertex i, and equal to zero on all other vertices.

The variables in the MHD equations are represented as a sum over basis functions. We first consider discretization of the equations (7), (8), (9), and (10), in which the variables to be expanded in basis functions are the velocity stream function ϕ, the magnetic flux ψ, vorticity Ω, and the current C. The variables in the MHD equations, such as ψ, are represented as

$$\psi(\mathbf{r}, t) = \sum_i \psi_i(t) \lambda_i(\mathbf{r}). \tag{16}$$

We use a zero residual Galerkin approach in which the equations are multiplied by a basis function λ_j, and integrated over the domain. This gives the set of sparse matrix equations,

$$\begin{aligned}
\mathbf{M} \cdot \frac{\partial}{\partial t}\mathbf{\Omega} + \mathbf{P} : \mathbf{\Phi\Omega} &= \mathbf{P} : \mathbf{\Psi C} + \mu \mathbf{S} \cdot \mathbf{\Omega} \quad , &(17)\\
\mathbf{M} \cdot \frac{\partial}{\partial t}\mathbf{\Psi} + \mathbf{P} : \mathbf{\Phi\Psi} &= 0. &(18)\\
\mathbf{S} \cdot \mathbf{\Phi} &= \mathbf{M} \cdot \mathbf{\Omega} &(19)\\
\mathbf{M} \cdot \mathbf{C} &= \mathbf{S} \cdot \mathbf{\Psi} &(20)
\end{aligned}$$

where

$$\begin{aligned}
(\mathbf{M} \cdot \mathbf{C})_i &= \sum_j M_{ij} C_j &(21)\\
(\mathbf{P} : \mathbf{\Phi\Psi})_i &= \sum_{jk} P_{ijk} \phi_j \psi_k &(22)
\end{aligned}$$

The matrices appearing in these equations are the mass matrix $\mathbf{M}$, the stiffness matrix $\mathbf{S}$, and the Poisson bracket tensor $\mathbf{P}$, defined by

$$\begin{aligned} M_{ij} &= \int \lambda_i \lambda_j d^2x & (23) \\ S_{ij} &= -\int \nabla\lambda_i \cdot \nabla\lambda_j d^2x & (24) \\ P_{ijk} &= \int \lambda_i [\ \lambda_j,\ \lambda_k\ ]\, d^2x & (25) \end{aligned}$$

Both the stiffness and mass matrices are symmetric. The Poisson bracket is anti symmetric under the exchange of any two indices. This assures that some of the most important integral relations satisfied by the differential equations are preserved by the finite element discretization. This includes conservation of energy and magnetic flux in the absence of dissipation. The matrices are sparse, having nonzero elements only between those vertices connected by the side of a triangle.

The discretization of the current - vorticity forms of the MHD equations is similar and involves the same matrices, as well as the gradient matrix, which can readily be constructed from the Poisson bracket matrix, using the spatial coordinates x, y as one of the arguments,

$$\mathbf{D}_{ij} = \int \lambda_i \nabla\lambda_j d^2x = \sum_k P_{ijk}(y_k\hat{\mathbf{x}} - x_k\hat{\mathbf{y}}). \qquad (26)$$

The gradient matrix is used to discretize the partial derivatives of ψ and ϕ in the Poisson brackets on the right side of (12). It is also used to develop a more accurate stiffness matrix, derived below in (36).

5 Triangle Based Matrices

The matrices appearing in (23), (24), and (25) can be calculated analytically in each triangle and accumulated on the triangle vertices. The matrices have a simple form locally in each triangle. In each triangle, introduce a local numbering of the vertices $\mathbf{r}_\alpha$, $\alpha = 1, 2, 3$, labeled in a counter clockwise order. The side segments $d\mathbf{r}_\alpha$ are labeled by their opposite vertex,

$$d\mathbf{r}_1 = \mathbf{r}_2 - \mathbf{r}_3,$$

and so on in cyclic order, or

$$d\mathbf{r}_\alpha = \frac{1}{2}\sum_{\beta\gamma} \epsilon_{\alpha\beta\gamma}(\mathbf{r}_\beta - \mathbf{r}_\gamma), \tag{27}$$

where the permutation symbol $\epsilon_{\alpha\beta\gamma} = 1$ if the values of the indices are in ascending order or a cyclic permutation, $= -1$, if the indices are in descending order or a cyclic permutation, or $= 0$, if any two indices have the same value.

The triangle area Δ is given by

$$\Delta = \frac{1}{2} d\mathbf{r}_1 \times d\mathbf{r}_2 \cdot \hat{z} = \frac{1}{4}\sum_{\beta\gamma} \epsilon_{\alpha\beta\gamma} d\mathbf{r}_\beta \times d\mathbf{r}_\gamma \cdot \hat{z} \tag{28}$$

The basis functions can be expressed as

$$\lambda_\alpha(\mathbf{r}) = \frac{1}{4\Delta}\sum_{\beta\neq\alpha} (\mathbf{r} - \mathbf{r}_\beta) \times d\mathbf{r}_\alpha \cdot \hat{z}. \tag{29}$$

To calculate the matrices, we need integrals of basis functions over the triangle, which are

$$\int_\Delta \lambda_1^\ell \lambda_2^m \lambda_3^n d^2x = 2\Delta \frac{\ell! m! n!}{(\ell + m + n + 2)!}$$

The contribution to the mass matrix from a single triangle, $\tilde{\mathbf{M}}$ is

$$\tilde{M}_{\alpha\beta} = \begin{cases} \Delta/6, & \alpha = \beta \\ \Delta/12, & \alpha \neq \beta \end{cases} \tag{30}$$

To calculate the stiffness matrix and Poisson bracket tensor, we need the gradients of the basis functions, which are constant in each triangle, and from (29) are given by

$$\nabla\lambda_\alpha = \frac{d\mathbf{r}_\alpha \times \hat{z}}{2\Delta}. \tag{31}$$

This immediately yields the contribution to the stiffness matrix from a single triangle, $\tilde{\mathbf{S}}$,

$$\tilde{S}_{\alpha\beta} = -\frac{d\mathbf{r}_\alpha \cdot d\mathbf{r}_\beta}{4\Delta}. \tag{32}$$

The contribution in a triangle to the Poisson bracket tensor is given by $\tilde{\mathbf{P}}$,

$$\tilde{P}_{\alpha\beta\gamma} = \int_\Delta \lambda_\alpha [\ \lambda_\beta,\ \lambda_\gamma\ ]\ d^2x = \frac{1}{12}\sum_\delta \epsilon_{\beta\gamma\delta} \tag{33}$$

which is independent of α.

The local matrices are assembled globally by summing the contribution of each triangle which shares a given vertex i.

It is convenient to consider a diagonalized form of the mass matrix, called the lumped mass matrix, formed by subtracting all the off diagonal matrix elements in each row and adding them to the diagonal,

$$m_i^\ell = \sum_k M_{ik} = \frac{1}{3} \sum_{a(i)} \Delta_a, \tag{34}$$

where $a(i)$ is the label of a triangle having vertex i. Its value, from (30), is one third the area of the triangles surrounding the vertex.

Using the lumped mass, the finite element discretization is equivalent to a finite volume discretization, where the control volumes are constructed by joining the barycenters of the triangles (average of the vertex positions) to the midpoints of the triangle edges.

6 Laplacian

Finite element representations are accurate and convergent in an integral sense, such as in L_2 norm. This does not mean, however, that the error at any given point will converge to zero as the mesh is refined. For this reason, the errors in the finite element Laplacian can be a source of noise, unless the mesh is uniform. This is not a problem when one wants to *invert* the Laplacian, as in eq.(19). This is a smoothing operation and the solution is quite acceptable. On the other hand, *calculating* the Laplacian, as in eq.(20) can be quite inaccurate locally.

Forming the Laplacian by multiplying by the stiffness matrix and the inverse mass matrix, $\mathbf{M}^{-1} \cdot \mathbf{S}$, yields excellent results on a uniform mesh. The Laplacian of a smooth function on this mesh is also smooth. Unfortunately, when the mesh is adaptively refined, it typically becomes non uniform. The mesh shown in Fig.1(a) has uniformly spaced mesh points, but the triangle vertices have alternately 4 and 8 neighbors. Fig.1(b) shows C, the discretized Laplacian of $\psi_i = \sin x_i \sin y_i$ on this mesh,

$$C(\mathbf{r}) = \sum_{ij} \frac{1}{m_i^\ell} S_{ij} \psi_j \lambda_i(\mathbf{r})$$

Refining the mesh is of no help, if it retains the alternating number of neighbors. This example shows that in some cases, the usual Laplacian is not locally convergent.

A uniform mesh in which all triangle vertices have 6 neighbors can be obtained from this mesh by a simple reconnection operation. One approach is to use vertex reconnection to try to equalize the number of neighbors of each vertex. However, with more complex meshes, it is virtually impossible to do this.

A way to improve the calculation of the Laplacian is to obtain the discretization of $\nabla^2\psi$ by applying a discretized gradient operator twice. First, the components of $\nabla\psi$ are calculated and re expanded in basis functions. Then, the discretized derivatives of these quantities are obtained to get $\nabla\cdot\nabla\psi$, which is again expanded in basis functions. The lumped mass is used to make the solution for the derivatives trivial. This method improves the accuracy because the $\nabla\psi$ components are first averaged from triangles to vertices, smoothing the gradient. The divergence calculation performs a similar smoothing. The method works because taking a single derivative is first order accurate. Taking a second derivative in two steps is formally first order accurate, and the method in practice is found to be convergent, unlike the original method. The discretized version of the gradient of a function ψ is

$$(\nabla\psi) = \sum_{i,j,k} M_{ij}^{-1}\mathbf{D}_{jk}\psi_k\lambda_i(x,y) \tag{35}$$

where $\mathbf{D}$ is the gradient matrix, (26).

Taking the divergence of (35), multiplying by a basis function, and integrating over the domain is equivalent to introducing a new stiffness matrix, S', given by

$$S'_{ij} = \sum_{k,l} M_{kl}^{-1}\mathbf{D}_{ik}\cdot\mathbf{D}_{lj} \tag{36}$$

In the above eqs. (35) and (36), it is sufficient to use the lumped mass matrix (34). This stiffness matrix has the same symmetry properties as the standard stiffness matrix. Fig.1(c) was obtained using the standard lumped mass and $\mathbf{S}'$.

Although the Laplacian has much better accuracy using this method, there is some loss of numerical stability, because the effective stencil of the Laplacian is now larger.

For adaptive problems, the best method appears to be the use of the current - vorticity formulation. In this approach, the Laplacian does not appear at all on the right hand side of the equations, only on the left hand sides of the two Poisson equations for the potentials. There appears to be no problem with smoothness of the potentials, using the original stiffness matrix, even on an irregular mesh. This because, as remarked above, the inversion of the Poisson equation is a smoothing operation. Both the currents and potentials have sufficient smoothness to obtain acceptable solutions.

7 Algorithm

We have written a two dimensional, adaptive, finite element MHD code, called FEMHD. The code solves the equations (7), (9), and (8), (10) or (12), (13). The code is written in modular form. One module generates the mesh and various arrays which specify its connectivity. The fundamental mesh arrays are the lists of N vertices $\mathbf{r}_i$, and an array which lists the three vertex indices of each of the triangles.

From these arrays, useful secondary arrays are constructed. These arrays list the neighboring vertices of each vertex, which are connected by a triangle side. Also listed are the number of neighbors, as well as the triangle numbers of the triangles sharing a particular vertex.

Another module contains routines for constructing and assembling the mass and stiffness matrices, the Poisson bracket tensor, and for inverting sparse symmetric matrices. We invert the stiffness and mass matrices using a sparse matrix Conjugate Gradient algorithm with an Incomplete Cholesky preconditioner.

The discretized equations are solved in a separate module. The time derivatives are represented by second order finite differences. The equations are advanced explicitly, using a leapfrog time differencing, but with the dissipative terms in the equations (7), (8) solved implicitly, by combining them with the mass matrix.

8 Adaptive Refinement

Mesh refinement is especially important, being one of the major reasons for using an unstructured mesh. Adaptive methods are particularly useful when there are isolated current sheets, as in the following, although less useful for fully developed turbulence.

The mesh refinement method works especially well for right triangles. The splitting operation is reversible and involves pairs of triangles. To be eligible for splitting, a pair must have their common side opposite their largest interior angles. This ensures, in the case of splitting, that the new triangles will not have excessively acute interior angles.

The mesh is refined by splitting pairs of triangles into four triangles. A new vertex is added at the midpoint of the common side. If the triangles are right triangles, only pairs with a common hypotenuse can be split. Isoceles right triangles will be divided into isoceles right triangles. They can be subdivided arbitrarily many times by this method. Non isoceles right triangles will generate geometrically similar descendants after two splittings, and can also be subdivided indefinitely. When a vertex is added, all MHD variables are linearly interpolated onto the new vertex, consistent with the piecewise linear representation.

The mesh is unrefined by reversing the division process. Four triangles with a common vertex can be unsplit into two triangles by removing the vertex. A diagonal is drawn across the four remaining vertices.

We have an adaptive algorithm to refine the mesh as the computation proceeds. Because we want to resolve current sheets, we monitor the current. If the product of current density times triangle area, $C\Delta$, exceeds a threshold, we split the triangle in two. The local current density in a current sheet typically rises exponentially in time, so the refinement process has this same behavior. We have to stop the refinement at some preset triangle size and number of mesh points. Similarly, if the current is too low, the triangles at that vertex are unrefined. There are of course other possible refinement criteria [10, 11].

The justification for using the product of current density and triangle area as a refinement criterion, is that $C\Delta$ is comparable to the discretization error in the magnetic flux ψ. It is very important to conserve magnetic flux in the neighborhood of a current sheet, in order to prevent numerical reconnection, as will be shown in the next section.

It is desirable to make $C\Delta$ as small as possible near a current sheet. Elsewhere, $C\Delta$ can be larger, to avoid unnecessary mesh refinement. A way to do this is to weight the refinement criterion. One possible weighting, which was used in the following, takes advantage of the fact that current sheets form on the magnetic separatrix. The separatrix is the locus of points having a particular ψ value, ψ_s, which divides different flux regions. The refinement criterion was

$$C\Delta > c_0 C_0 \Delta_0 [1 + (\frac{\psi - \psi_s}{\delta\psi_0})^2]. \tag{37}$$

where c_0, δ are constants, and ψ_0, C_0, and Δ_0 are reference maximum values of those respective quatities. The criterion is calculated using the average of C and ψ on each triangle. For the maximum localization, the constant δ can be set as small as possible, but not so small that the bracketed term on the right side of (37) is large on mesh points of triangles which cross the separatrix, on the unrefined mesh. This has an effect similar to increasing c_0.

9 Computational Results: Coalescence Instability

As an example and test of the method, we first consider the periodic coalescence instability [1, 2].

The initial equilibrium for the periodic coalescence instability consists of an array of cells with

$$\psi = A\left[\cos(4\pi x/\ell) - \cos(4\pi y/\ell)\right], \tag{38}$$

$C = -(4\pi/\ell)^2\psi$, and $\phi = \Omega = 0$. We choose the constants $A = 1, \ell = 1$. The initial equilibrium flux function ψ is shown in Fig.3(a).

We solve the time dependent equations in the domain $0 \le x \le \ell$, $0 \le y \le \ell$, with periodic boundary conditions. Periodicity is built into the mesh by the connectivity of the mesh triangles. With periodic boundary conditions, coordinate differences are calculated modulo ℓ to obtain the mesh - dependent mass and stiffness matrices.

The initial equilibrium is unstable to small perturbations. Simulations were initialized with arbitrary velocity perturbations, from which the unstable mode grows as $\exp\gamma t$. The growth rate was extracted by monitoring the kinetic energy as a function of time. Runs were made advancing both the potential and current advection forms of the equations, for several initial mesh sizes. No adaptation was done for these linear computations. Fig.2 shows the growth rate γ as a function of number of mesh points N. The two upper curves show $\gamma(N)$ for zero viscosity, using the two forms of the equations. The curve made with the current form is marked with x's, while the curve made with the potential advance using the modified Laplacian is marked with o's. There is little difference in the results, which appear to asymptote to the same linear growth rate. Also shown is a dashed line, which is the growth rate with a viscous term, with $\mu = 0.025$, corresponding to the value of μ in finite difference viscous simulations [2]. The linear growth rate is in excellent agreement with the viscous finite difference simulations. For a mesh of size 100×100, the finite difference growth rate $\gamma = 42$. Here, the growth rate for $N = 10,000$ is (by interpolation) $\gamma = 41$.

We now consider an adaptive computation of the coalescence instability. A small viscosity of $\mu = 0.005$ is used. We use the current - vorticity advection formulation of the MHD equations. Starting with an initial grid of $N = 2500$ points, the code evolves the equations and refines the mesh. The flux function ψ at time $t = 0.23$ is shown in Fig.3(b). The contours of ψ have the form of cells divided by a nearly pentagonal separatrix. In [1] it was shown that there is an equilibrium, with pentagonal separatrices, which has a singular current density along the shortest side of each pentagon. This equilibrium has lower energy, and conserves magnetic flux, relative to the initial state. The singular equilibrium might be expected as the final state of the time dependent evolution.

As the simulation evolves, the current density becomes concentrated into thin sheets located at the short side of the pentagonal separatrix. A blowup of the plot of the current density at time $t = 0.23$ is shown in Fig.4(a). A cross sectional plot of the current density is shown in Fig.4(b). The current is well resolved and unremarkable in structure. A similar blowup of the mesh on which the current is calculated is shown in Fig.5. The minimum length scale of the mesh is .00388 the length of the original mesh cells, which is equivalent to a mesh of $166,334,000$ mesh points. In fact the mesh has $87,770$ mesh points.

The peak value of the current density grows exponentially in time, with a large growth rate more than 10 times the linear mode growth rate. The logarithm of the peak current density grows approximately linearly. Exponential growth is predicted theoretically [6]. As the current density increases, so does the number N of mesh points. The growth of the current density is insensitive to the refinement criterion for early times, but progressively more refinement is needed to continue the computation in time. A comparison of the peak current density as a function of time, with different refinement criteria (37), is shown in Fig.6(a). The runs were done with $c_0 = .005, .045$, and with no refinement. Both adaptive runs had $\delta = 0.02$, approximately the minimum value. The upper curve, with $c_0 = .005$, corresponds to the run shown in the previous figure. The middle curve, with $c_0 = .045$, at first is the same as the upper curve, but then rises more slowly. The lowest curve, with no refinement, has the same growth at first, but does not reach the peak values of the other curves. As the current density increases, so does the number of mesh points, as shown in Fig.6(b). The high resolution run has more than an order of magnitude more mesh points than the medium resolution run, which has 7952 mesh points.

In all three cases, the total kinetic and magnetic energy is conserved within 0.2%. The most important difference between the runs is the conservation of magnetic flux at the current sheet, and the amount of numerical reconnection. The ψ contours at $t = .23$, for the unadapted case, are shown in Fig.7(a), which should be compared with Fig.3(b). The originally diamond shaped flux regions (islands) are merging together, a process not allowed in ideal MHD. This can be quantified by considering the magnetic flux ψ_s measured between the merging flux regions in the center of the figure, and ψ_0, the maximum value of the flux, measured in the center of the islands. The flux difference,

$$f = 1 - \frac{\psi_s}{\psi_0}$$

is plotted as a function of time in Fig.7(b). The upper curve is the high resolution case, and the lower curve is the unadapted case. The moderately refined case is similar to the high resolution case, but has somewhat larger fluctuations. Even for the highest resolution, there is some reconnection, but far less than without mesh refinement.

10 Computational Results: Tilt Mode

We next consider the two dimensional tilt instability [5]. In this calculation we have used the lumped mass matrix and the current - vorticity formulation of MHD.

The initial equilibrium state is a bipolar vortex,

$$\psi = \begin{cases} [2/kJ_0(k)]J_1(kr)\cos\theta, & r < 1, \\ (1/r - r)\cos\theta, & r > 1. \end{cases} \tag{39}$$

$$J_1(k) = 0$$

When perturbed, an instability occurs, growing exponentially as $\exp\gamma t$.

We perform a simulation with an initial mesh, with 40 triangles on a side. Starting with the equilibrium of eq.(39), shown in Fig.8(a), a perturbation about 10^{-3} smaller is inserted. In the simulation, we take $\mu = 0.005$, and the simulation box has sides of length 4. Conducting boundary conditions are applied on the walls, at which $\phi = 0$, and $\partial\psi/\partial t = 0$.

The previous simulations [5] were compressible, and growth rates were reported in the range $1.3 < \gamma < 1.4$, depending on the pressure. None of these cases are exactly equivalent to our strictly incompressible model. We obtain the linear growth rate $\gamma = 1.2$.

Adaptive simulations were done with the current advection scheme. In the simulation, the motion is highly nonlinear by time $t = 7$. The initial ψ is shown in Fig.8(a), and ψ at time $t = 7$, in the nonlinear stage, is shown in Fig.8(b). At this stage, the vortex has tipped over. The separatrix wraps around the two flux vortices. Current sheets are formed at the leading edges of the central vortices, which can be seen in a blowup view in Fig.9(a). The mesh supporting the contours is shown in the same blowup view Fig.9(b). The mesh resolution has adaptively followed the formation of the moving, curved current sheet. The peak value of the current density grows exponentially in time, with a large growth rate about 3 times the linear mode growth rate. This can be seen in Fig.10(a), which shows the logarithm of the peak current density as a function of time. The logarithm of the peak current density grows approximately linearly. The two curves are from runs with different refinement parameters (37). The upper curve, as well as the plots of Fig.8, Fig.9, were done with a threshhold $c_0 = .05$, and the lower curve with $c_0 = .15$. There was relatively little difference in the results. In both cases $\delta = 0.15$, which gave relatively less localization of the refinement about the separatrix than in the coalescence simulations. As the current density increases, so does the number N of mesh points, shown in Fig.10(b). Again, the upper curve had the lower refinement threshhold, which caused more refinement both in the initial and latter stages of the simulation. The runs ended with $73,700$ and $23,670$ mesh points, respectively.

11 Other Applications

The unstructured mesh methods described above have been also applied in problems with a complicated boundary shape.

In [7] the equations were supplemented by additional equations for variables p and v; this system is known as Compressional Reduced MHD [7, 8]. The equations were solved on the computational mesh shown, at a low resolution of 800 mesh points for clarity, in Fig.11(a). The contours of the equilibrium magnetic flux ψ are shown in Fig.11(b).

In [8] the equations were solved in three dimensions, in the CRMHD approximation, where the third dimension was discretized by finite differences. The grid in the

x, y plane was independent of the third coordinate. The x, y grid was similar to that of Fig.11(a).

Finally the finite element discretization described here is being combined with an existing 3D full MHD code to give a highly flexible and powerful method for solving 3D nonlinear MHD problems in complex geometry [9]. Again the x, y grid is independent of the third coordinate, which is discretized using Fourier series.

12 Conclusion

MHD motion tends to produce nearly discontinuous magnetic fields and singular current density. In general, an adaptive numerical method is needed. We have solved two dimensional, incompressible MHD equations on an unstructured grid of triangles, using a a piecewise linear finite element discretization. A stream function representation of the magnetic and velocity fields is used to ensure zero divergence of the fields.

A straightforward application of this approach causes problems with the current, due to the local non convergence of the finite element Laplacian. An improved Laplacian can be constructed by successive application of the finite element gradient and divergence. For adaptive computations, the best results are obtained by reformulating the MHD equations so that the vorticity and current are time advanced, with the magnetic and velocity stream functions found by solving Poisson equations.

For adaptive computations, mesh operations are provided to reconnect triangles and to refine (and unrefine) the mesh. The mesh adaptively refines to resolve current sheets, while conserving magnetic flux. Example simulations of the coalescence and tilt instability show the formation of current sheets, with the current density increasing exponentially in time.

The unstructured mesh methods described in this paper have also been applied to problems in which the computational boundary has a complicated shape. The methods are being incorporated in a three dimensional MHD code, in which the mesh is unstructured in two dimensions, and structured in the third dimension.

Acknowledgment

We wish to thank Anne Greenbaum of NYU, who provided us with a sparse matrix ICCG routine. This work was supported by AFOSR Grant No. 91-0044 and U. S.

Department of Energy Grant DE-FG02-86ER53223. Some of the computations were performed at NERSC, operated by the USDOE.

References

[1] D. W. Longcope and H. R. Strauss, Phys. Fluids B5, 2858 (1993).

[2] D. W. Longcope and H. R. Strauss, Spontaneous reconnection of line tied flux tubes, Ap. J. 426:742-757, (1994).

[3] J. U. Brackbill, *Methods in Computational Physics*, edited by J. Kileen (Academic Press, New York, 1976), Vol. 16, p. 1.

[4] W. Park, D. A. Monticello, R. B. White, and S. C. Jardin, *Nuclear Fusion***20**, 1181 (1980).

[5] R. L. Richard, R. D. Sydora, and M. Ashour - Abdalla, Phys. Fluids **B 2**, 488, 1990.

[6] I. Klapper, Constraints on finite - time current sheet formation at null points in two - dimensional ideal incompressible magnetohydrodynamics, Physics of Plasmas, *to appear*, (1998).

[7] H. R. Strauss, Physics of Plasmas *2*, 1229 - 1235, (1995).

[8] H. R. Strauss, Edge - localized mode simulations in divertor tokamaks, Phys. Plasmas *3*, 4095 (1996).

[9] W. Park, Z. Chang, E. Fredrickson, G. Y. Fu, N. Pomphrey, H. R. Strauss, and L. E. Sugiyama, 3D Simulation Studies of Tokamak Plasmas using MHD and Extended - MHD Models, Sixteenth IAEA Fusion Energy Conference, Montreal, F1-CN-64/D2-2 (1996)

[10] H. Friedel, R. Grauer and C. Marliani, Adaptive mesh refinement for singular current sheets in incompressible magnetohydrodynamic flows, J. Comp. Phys. **134**, 190 - 198 (1997).

[11] D. D. Schnack, I. Lottati, Z. Mikić, and P. Satyanarayana, A finite volume algorithm for three dimensional magnetohydrodynamics on an unstructured, adaptive grid in axially symmetric geometry, J. Comp. Phys., to appear, 1998.

13 Figure Captions

Fig.1 (a) Effect of mesh non - uniformity on Laplacian: a mesh with uniform mesh spacing but non - constant number of neighbors of each triangle vertex.

(b) Laplacian of $\sin \pi x \sin \pi y$ on mesh (a), using standard stiffness matrix;

(c) Laplacian of $\sin \pi x \sin \pi y$ on mesh (a), using modified stiffness matrix, eq.36.

Fig.2 Growth rate γ as a function of number of mesh points N. The upper curves were obtained for zero viscosity, by advancing the magnetic potential or the current. The current advancement method gives somewhat larger γ for low N, but the two methods agree closely for larger N. The lower curve was obtained with viscosity $\mu = 0.025$, which agrees with a previously reported, finite difference calculation.

Fig.3 (a) A calculation of the coalescence instability. Contours of magnetic flux ψ at time $t = 0$. The flux forms a checkerboard diamond shaped pattern. The contours are equally spaced in the range $-2 < \psi < 2$.

(b) Contours of magnetic flux ψ at time $t = 0.23$ The diamonds have distorted to form pentagons. Current sheets have formed along the short sides of the pentagons.

Fig.4 (a) A blowup view of contours of current C at time $t = 0.23$ The view is centered on the separatrix, on the short side of a flux pentagon. The horizontal scale is about 1/4 of the scale l of the previous figure, while the vertical scale is about $0.02l$. The current sheet has unremarkable structure.

(b) A line plot of the current density in a vertical slice through the middle of the current sheet.

Fig.5 A blowup view of the mesh supporting the previous figure. The mesh is highly refined along the current sheet, which is well resolved. The minimum scale length of the mesh is 0.00388 the size of the initial mesh separations.

Fig.6 (a) Time history of the log of the peak current density. In the latter part of the run, the current density grows exponentially. The upper curve has the most mesh refinement, the middle curve has moderate mesh refinement, and the lower curve has no mesh refinement.

(b) Time history of the number of mesh points N. In the latter part of the run, the number of mesh points grows exponentially, keeping pace with the peak current density. Again, the upper curve has the most mesh refinement, the middle curve has moderate mesh refinement, and the lower curve has no mesh refinement.

Fig.7 (a) Contours of magnetic flux ψ at time $t = 0.23$ with no mesh refinement. Because of numerical dissipation, the flux regions merge. Reconnection is not allowed in ideal MHD. (b) Time history of the flux reconnection parameter f. The upper curve, for the high resolution case, shows little reconnection, while the lower curve, for the unadapted run, shows about 60% of the flux has reconnected.

Fig.8 (a) A calculation of the bipolar vortex tilt instability. Contours of magnetic flux ψ at time $t = 0$. The flux contours consist of two interior flux systems, centered on o - points, and an exterior flux system whose contours intersect the boundary. The contours are equally spaced in the range $-1.8 < \psi < 1.8$.

(b) Contours of magnetic flux ψ at time $t = 7$. The inner flux blobs have tilted from their initial positions. The separatrix winds around the edges of the tilted flux blobs.

Fig.9 (a) A blowup view of contours of current C at time $t = 7$. The current is localized along the separatrix.

(b) A blowup view of the mesh, corresponding to the contours in (a). The refinement is able to resolve the moving, curved current sheet.

Fig.10 (a) Time history of the log of the peak current density, for two values of refinement threshhold. The upper curve has the lower threshhold. In the latter part of the run, the current density grows exponentially, as indicated by the approximately linear growth of the log of the peak current.

(b) Time history of the number of mesh points N. The upper curve has the lower threshhold. In the latter part of the run, the number of mesh points grows exponentially, keeping pace with the peak current density.

Fig.11 (a) Mesh used for magnetic separatrix computation. There are about 800 mesh points.

(b) Equilibrium magnetic flux function ψ calculated with 2D CRMHD, on a mesh like that of (a), but with about four times as many mesh points.