FREE SURFACE ELECTROMAGNETICALLY RESTRAINED LIQUID LITHIUM BLANKET STATUS

BY
ROBERT D. WOOLLEY

APEX MEETING
UCLA
8-11 NOVEMBER 1999

Background

PROPOSED FUSION REACTOR BLANKET CONCEPT:

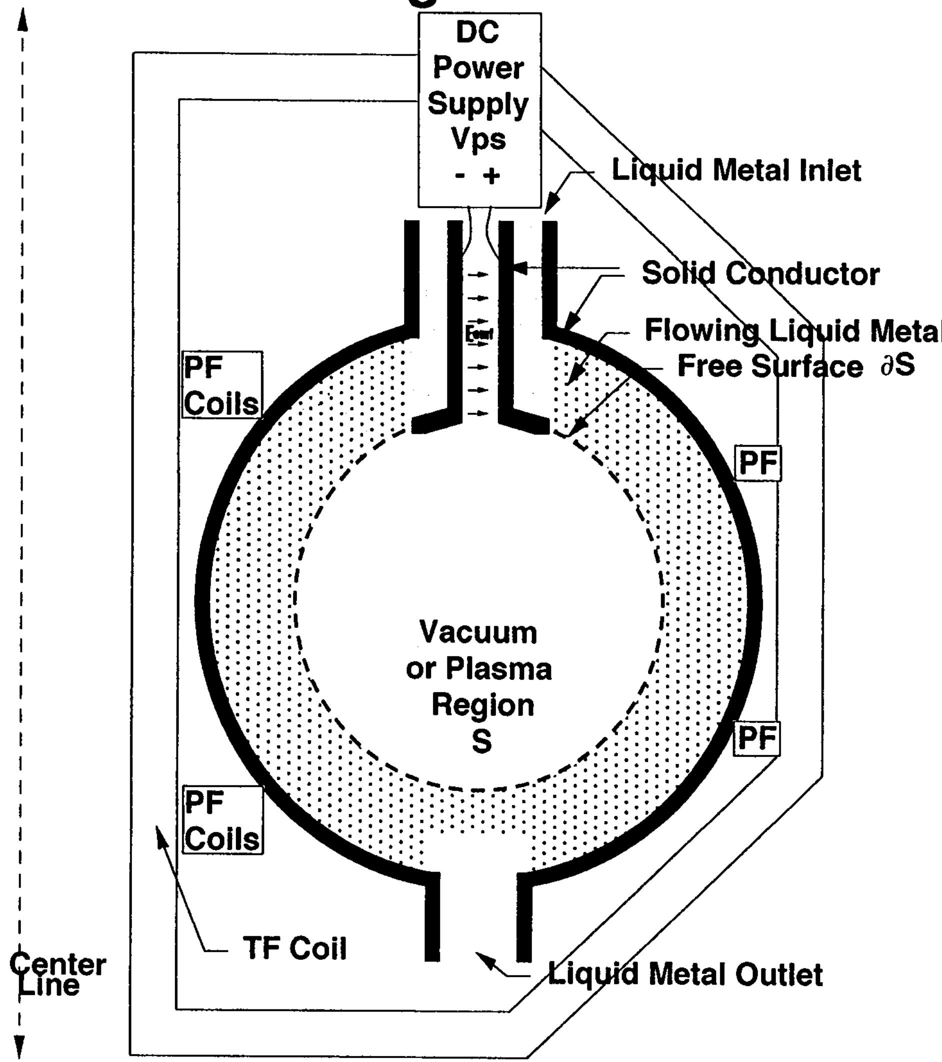
CONFINE A THICK (1+ METER) FLOWING LIQUID LITHIUM LAYER BY ELECTROMAGNETIC, CENTRIFUGAL, AND CONTACT FORCES TO ENCLOSE A TOROIDAL MAGNETICALLY CONFINED PLASMA.

Two axisymmetric liquid lithium streams enter toroidal chamber's top. The two streams are electrically separated at the top, biased to different voltages via electrodes connected to an external power supply. (Some leakage current may flow through non-insulating structure.) Poloidal current injected via these electrodes is driven through the streams which meet at the bottom of the chamber.

The resulting J X B forces push the streams against the chamber walls away from the plasma.

This is analogous to an extra TF coil turn, but of flowing liquid lithium.

General Electromagnetic Restraint Scheme



Liquid Walls

A sufficiently thick, flowing, liquid first wall and tritium breeding blanket which almost completely surrounds a fusioning tokamak DT plasma could provide superior features in a commercial MFE powerplant.

- Much of the costly complexity of conventional solid first wall designs would be eliminated.

- It would eliminate first wall damage from neutrons, from high heat flux, and from disruptions.

- Power density would not be limited by damage issues, so higher power density may be possible.

- It would eliminate solid surface tritium inventories which are not accessible.

- If sufficiently thick (> 1 m lithium), it would eliminate the need to develop exotic structural materials more resistant to neutron damage.

- If sufficiently thick (> 1 m lithium), it would reduce the activation and waste disposal requirements for structural materials.

Liquid Walls

A nonconducting or poorly conducting liquid material containing lithium such as the molten salt, FLiBe, could provide a liquid first wall&breeding blanket with minimal MHD interactions. Centrifugal force would need to be used to keep it out of the plasma. However, use of molten lithium could also bring other benefits:

- Lithium is near-ZERO activation material, and is not especially toxic.

- Lithium is low-Z (after hydrogen and helium) and so could be expected to have small impact on plasma radiation losses.

- Experience and some theory suggest lithium impurities may improve plasma performance.

- Lithium walls could reduce vacuum pumping problems.

- Use of a single element may simplify chemical processing of the blanket.

- Lithium could also be used for direct LMMHD production of electricity, using the TF magnets as the LMMHD power production magnets, further reducing capital costs for a fusion power plant.

Electromagnetic Issues

•FLOW NEEDS TO FOLLOW POLOIDAL FLUX

If liquid metal flow in a poloidal plane cuts across poloidal flux lines, the resulting V X B electric field will produce a toroidal current locally in the liquid metal, which in turn will modify the poloidal field needed for plasma equilibrium. It may be necessary to design the LM flow path following poloidal flux lines near the plasma's last closed flux surface.

•DIAMAGNETIC DRAG

The radial LM velocity through the toroidal field introduces a diamagnetic drag force opposing that motion. The effect depends on how rapidly the flux imbalance diffuses away. The parameter, $\lambda = 1/\mu_0\sigma = 0.36$ s m^{-2}, predicts magnetic diffusion through 1.7 meters of lithium in one second.

INJECTED POLOIDAL CURRENT.

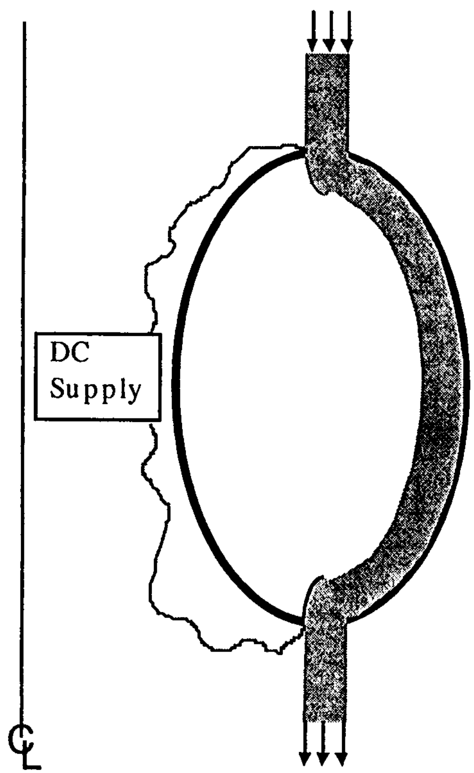
The injected poloidal current needs to follow the poloidal field direction in order to avoid creating a toroidal swirl motion in the liquid metal.

FREE SURFACE SHAPE

The free surface shape results from the interplay of flow parameters in a way not yet known. It is expected sensitive to injected current strength, injected fluid speed, and shape of backing surface.

SIMULATION NEED

For engineering investigations of candidate AXISYMMETRIC, FREE SURFACE flows, a numerical simulation computer code is needed. No preexisting free-surface LMMHD code was found, so a new code has been developed. Its general objectives are as follows:



It should allow changing
- -vessel and electrode geometry
- -liquid inflow rate & pressure
- -electrode voltage

It should calculate free surface shape and the distributions of:
- -electrical currents in the lithium and vessel wall
- -magnetic field
- -the liquid lithium's velocity field
- -the liquid lithium's pressure field
- -transit time, especially at the free surface

Later, it should model a plasma.

PHYSICS: GENERAL VECTOR EQUATIONS & BCs ARE COMPLEX

Incompressible	$\nabla \bullet \vec{\mathbf{u}} = 0$
Navier-Stokes:	$\frac{\partial \vec{\mathbf{u}}}{\partial t} + (\vec{\mathbf{u}} \bullet \nabla)\vec{\mathbf{u}} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \vec{\mathbf{u}} + \vec{\mathbf{g}} + \frac{\vec{\mathbf{J}} \times \vec{\mathbf{B}}}{\rho}$
Liquid at Free Surface:	$p = 0$
Liquid at Material Walls:	$p > 0; \quad \vec{\mathbf{u}} = 0$
Ampere:	$\nabla \times \vec{\mathbf{B}} = \mu \vec{\mathbf{J}}$
Kirchoff:	$\Rightarrow \nabla \bullet \vec{\mathbf{J}} = 0$
	$\nabla \bullet \vec{\mathbf{B}} = 0$
Faraday:	$\nabla \times \vec{\mathbf{E}} = -\frac{\partial \vec{\mathbf{B}}}{\partial t}$
Ohm:	$\vec{\mathbf{J}} = \sigma(\vec{\mathbf{E}} + \vec{\mathbf{u}} \times \vec{\mathbf{B}})$
Vector Potential	$\nabla \times \vec{\mathbf{A}} = \vec{\mathbf{B}}$
Coulomb Gauge	$\nabla \bullet \vec{\mathbf{A}} = 0$
Induction & Electric Field:	$\vec{\mathbf{E}} = -\nabla V - \frac{\partial \vec{\mathbf{A}}}{\partial t}$
Electrode Voltages:	$V_1 = 0; \quad V_2 = V_{PS}$
At infinite distance:	$\vec{\mathbf{A}}, \vec{\mathbf{B}} \rightarrow 0$

Note: Driven currents in TF and PF electromagnets influence $\vec{\mathbf{A}}, \vec{\mathbf{B}}$.

DERIVED LMMHD AXISYMMETRIC MATHEMATICAL FORMULATION

Because of the impressive computational demands of computing LMMHD flows, we do not want the computer to waste time and effort calculating nonuniform electromagnetic fields in vacuum regions.

We therefore seek a formulation in which the computational grid covers only the electrically conducting region (including liquid and solid metal).

As done in tokamak plasma physics MHD theory, we express the magnetic field via two scalar field variable functions of position and time:

$I(r,z,t)$ is the "poloidal threading current", the total net poloidal current between the center-line axis of symmetry and any (r,z) location.

$\Psi(r,z,t)$ is the "poloidal magnetic flux", the total net poloidal magnetic flux between the center-line axis of symmetry and any (r,z) location.

I has the distinction that it is spatially constant in vacuum regions.

Ψ has a simple enough form in vacuum regions that a Greens function method can be used at vacuum/conducting region boundaries.

Current and Magnetic Field In This Formulation

In terms of the two scalar variables, the total magnetic field is

$$\vec{B} = B_r \hat{a}_r + B_\varphi \hat{\varphi} + B_z \hat{a}_z = \frac{\mu I}{2\pi r}\hat{\varphi} + \nabla \times \left(\frac{\Psi}{2\pi r}\hat{\varphi}\right)$$

and the total current density is

$$\vec{J} = J_r \hat{a}_r + J_\varphi \hat{\varphi} + J_z \hat{a}_z = \frac{1}{\mu}\nabla \times \vec{B} = \nabla \times \left(\frac{I}{2\pi r}\hat{\varphi}\right) - \vec{\nabla}^2\left(\frac{\Psi}{2\pi r \mu}\hat{\varphi}\right)$$

$$= \nabla \times \left(\frac{I}{2\pi r}\hat{\varphi}\right) - \frac{\Delta^* \Psi}{2\pi r \mu}\hat{\varphi}$$

where Δ^* is the Grad-Shafranov operator, $\Delta^* := r\frac{\partial}{\partial r}\frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2}$

NAVIER-STOKES EQN. $\frac{\partial \vec{u}}{\partial t} + (\vec{u} \bullet \nabla)\vec{u} = -\frac{\nabla p}{\rho} + \nu \overleftrightarrow{\nabla}^2 \vec{u} + \vec{g} + \frac{\vec{J} \times \vec{B}}{\rho}$ becomes

$$\frac{\partial u_r}{\partial t} = \frac{u_\varphi^2}{r} - u_r \frac{\partial u_r}{\partial r} - u_z \frac{\partial u_r}{\partial z} - \frac{1}{\rho}\frac{\partial p}{\partial r} + \nu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r}{\partial r}\right) + \frac{\partial^2 u_r}{\partial z^2} - \frac{u_r}{r^2}\right)$$
$$- \frac{\Delta^* \Psi}{4\pi^2 r^2 \rho\mu}\frac{\partial \Psi}{\partial r} - \frac{\mu I}{4\pi^2 r^2 \rho}\frac{\partial I}{\partial r}$$

$$\frac{\partial u_\varphi}{\partial t} = -\frac{u_r u_\varphi}{r} - u_r \frac{\partial u_\varphi}{\partial r} - u_z \frac{\partial u_\varphi}{\partial z} + \nu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_\varphi}{\partial r}\right) + \frac{\partial^2 u_\varphi}{\partial z^2} - \frac{u_\varphi}{r^2}\right)$$
$$+ \frac{1}{4\pi^2 r^2 \rho}\left(\frac{\partial I}{\partial z}\frac{\partial \Psi}{\partial r} - \frac{\partial I}{\partial r}\frac{\partial \Psi}{\partial z}\right\}$$

If $\nabla\Psi \times \nabla I = 0$

then $u_\varphi(r, z, t) \equiv 0$

$$\frac{\partial u_z}{\partial t} = -u_r \frac{\partial u_z}{\partial r} - u_z \frac{\partial u_z}{\partial z} - \frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z}{\partial r}\right) + \frac{\partial^2 u_z}{\partial z^2}\right)$$
$$+ g_z - \frac{\Delta^* \Psi}{4\pi^2 r^2 \rho\mu}\frac{\partial \Psi}{\partial z} - \frac{\mu I}{4\pi^2 r^2 \rho}\frac{\partial I}{\partial z}$$

Pressure Poisson Equation (PPE)

Incompressibility requires that $\nabla \bullet \vec{u} = 0 \Rightarrow \nabla \bullet \frac{\partial \vec{u}}{\partial t} = 0$ **in the divergence of the N-S equation. Assuming constant properties, this gives**

$$\nabla^2 p = -\rho \nabla \bullet [(\vec{u} \bullet \nabla)\vec{u}] + \nabla \bullet (\vec{J} \times \vec{B})$$

In terms of this formulation the PPE can be rewritten as

$$\nabla^2 p = -\rho\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\left[\frac{u_\varphi^2}{r} - u_r \frac{\partial u_r}{\partial r} - u_z \frac{\partial u_r}{\partial z}\right]\right) + \frac{\partial}{\partial z}\left[-u_r \frac{\partial u_z}{\partial r} - u_z \frac{\partial u_z}{\partial z}\right]\right)$$
$$- \frac{1}{4\pi^2 r\rho}\left(\frac{\partial}{\partial r}\left(\frac{1}{r}\left[\frac{\Delta^* \Psi}{\mu}\frac{\partial \Psi}{\partial r} + \mu I \frac{\partial I}{\partial r}\right]\right) + \frac{\partial}{\partial z}\left(\frac{1}{r}\left[\frac{\Delta^* \Psi}{\mu}\frac{\partial \Psi}{\partial z} + \mu I \frac{\partial I}{\partial z}\right]\right)\right)$$

PPE boundary conditions constrain the normal derivative of pressure at material surfaces, derived from the need for compatibility with the N-S equation's zero perpendicular velocity boundary condition there. The PPE boundary condition on a free surface requires that absolute pressure balance surface tension, which depends on surface curvature.

Trapped Toroidal Flux Ordinary Differential Equation

It is necessary for the simulation to follow and predict the total toroidal flux enclosed by the liquid metal surface, since that trapped flux will be in a "computational hole" region. In terms of this formulation, its evolution equation is

$$\frac{d\Phi}{dt} = V_{PS} - \frac{1}{2\pi\sigma}\oint_{\partial S}\left(\frac{\partial I}{\partial r}\frac{dz}{r} - \frac{\partial I}{\partial z}\frac{dr}{r}\right) + \frac{\mu I_s}{2\pi}\oint_{\partial S}\frac{(u_r dz - u_z dr)}{r} + \frac{1}{2\pi}\oint_{\partial S}\frac{u_\varphi}{r}\,d\Psi$$

where V_{PS} is the (possibly time-varying) voltage of the galvanic power supply which drives the restraining currents in the liquid metal, I_S is the free-surface value of I, and ∂S denotes the free surface boundary over which these contour integrals are evaluated in the clockwise sense.

The first integral calculates the liquid metal's resistive voltage drop.

The second integral is only nonzero when the free surface boundary is moving in space.

The third integral is identically zero if either
the poloidal magnetic flux is constant on the free surface, or if
the toroidal "swirl" angular velocity is a constant on the free surface.

Poloidal Threading Current Evolution Equations

The poloidal threading current evolves in metal regions as

$$\frac{\partial I}{\partial t} = -u_r \frac{\partial I}{\partial r} - u_z \frac{\partial I}{\partial z} + \frac{1}{\sigma\mu}\Delta^* I + \frac{2u_r I}{r} + \frac{r}{\mu}\nabla\Psi \times \nabla\left(\frac{u_\varphi}{r}\right)$$

The first two terms on the right form a convective derivative. The next term is electrically resistive using a Grad-Shafranov operator. The cross product shows mismatch of toroidal "swirl" and poloidal field. The "drag" term, $\frac{2u_r I}{r}$, accompanies motion through field nonuniformity.

Between liquid and solid metal subregions the normal current and tangential electrical field are continuous. Thus for unit normal (n_r, n_z) ,

$$\left.\begin{bmatrix} \frac{n_r}{\sigma_{Liq}} & \frac{n_z}{\sigma_{Liq}} \\ n_z & -n_r \end{bmatrix}\begin{pmatrix} \frac{\partial I}{\partial r} \\ \frac{\partial I}{\partial z} \end{pmatrix}\right|_{Liquid} = \left.\begin{bmatrix} \frac{n_r}{\sigma_{Sol}} & \frac{n_z}{\sigma_{Sol}} \\ n_z & -n_r \end{bmatrix}\begin{pmatrix} \frac{\partial I}{\partial r} \\ \frac{\partial I}{\partial z} \end{pmatrix}\right|_{Solid}$$

The exterior boundary condition sets $I=I_{TF}$ to the TF coil system's ampere-turns. The inner (free) surface boundary condition is expressed via a free surface shape integral and Φ from the ODE, as $I=I_S$,

$$I_S = \frac{2\pi}{\mu} \frac{\Phi}{\oint_{\partial S} \frac{z}{r} dr}$$

Poloidal Magnetic Flux Evolution Equations & BCs

The poloidal magnetic flux evolves in metal regions as

$$\frac{\partial \Psi}{\partial t} = -u_r \frac{\partial \Psi}{\partial r} - u_z \frac{\partial \Psi}{\partial z} + \frac{1}{\sigma\mu} \Delta^* \Psi$$

Continuous tangential electric field between liquid and solid metal subregions requires

$$\left(\frac{\hat{n} \bullet \nabla \Psi_{Solid}}{\sigma_{Solid}} - \frac{\hat{n} \bullet \nabla \Psi_{Liquid}}{\sigma_{Liquid}} \right) \Bigg|_{boundary}$$

Greens functions can express the normal derivative $\frac{\partial \Psi}{\partial n}$ over the surface boundary contour as a linear transform of Ψ over the same contour, plus the sum of PF coil currents times associated boundary functions. For a finite number of boundary nodes this becomes a linear vector-matrix equation which replaces conventional Neuman or Dirichlet bcs and avoids computing Ψ through the vacuum region.

INITIAL SIMULATION-NO PF, PLASMA, TOROIDAL SWIRL

Setting $\Psi=0$ avoids plasma simulation, $\Rightarrow u_\varphi=0$

Poloidal Threading Current

$$\frac{\partial I}{\partial t} = -(u_r)\left(\frac{\partial I}{\partial r}\right) - (u_z)\left(\frac{\partial I}{\partial z}\right) + \frac{1}{\sigma\mu}\left[\left(\frac{\partial^2 I}{\partial r^2}\right) - \frac{1}{r}\left(\frac{\partial I}{\partial r}\right) + \left(\frac{\partial^2 I}{\partial z^2}\right)\right] + \frac{2}{r}(u_r)(I)$$

Radial Velocity Component

$$\frac{\partial u_r}{\partial t} = -(u_r)\left(\frac{\partial u_r}{\partial r}\right) - (u_z)\left(\frac{\partial u_r}{\partial z}\right) - \frac{1}{\rho}\left(\frac{\partial p}{\partial r}\right) + \nu\left[\left(\frac{\partial^2 u_r}{\partial r^2}\right) + \frac{1}{r}\left(\frac{\partial u_r}{\partial r}\right) - \frac{1}{r^2}(u_r) + \left(\frac{\partial^2 u_r}{\partial z^2}\right)\right] - \frac{\mu}{4\pi^2\rho r^2}(I)\left(\frac{\partial I}{\partial r}\right)$$

Vertical Velocity Component

$$\frac{\partial u_z}{\partial t} = -(u_r)\left(\frac{\partial u_z}{\partial r}\right) - (u_z)\left(\frac{\partial u_z}{\partial z}\right) - \frac{1}{\rho}\left(\frac{\partial p}{\partial z}\right) + \nu\left[\left(\frac{\partial^2 u_z}{\partial r^2}\right) + \frac{1}{r}\left(\frac{\partial u_z}{\partial r}\right) + \frac{\partial^2 u_z}{\partial z^2}\right] + g_z - \frac{\mu}{4\pi^2\rho r^2}(I)\left(\frac{\partial I}{\partial z}\right)$$

Pressure Poisson Equation

$$\frac{1}{\rho}\left(\frac{\partial^2 p}{\partial r^2}\right) + \frac{1}{\rho r}\left(\frac{\partial p}{\partial r}\right) + \frac{1}{\rho}\left(\frac{\partial^2 p}{\partial z^2}\right) =$$

$$-\frac{1}{r}(u_r)\left(\frac{\partial u_r}{\partial r}\right) - \frac{1}{r}(u_z)\left(\frac{\partial u_r}{\partial z}\right) - \left(\frac{\partial u_r}{\partial r}\right)^2 - 2\left(\frac{\partial u_z}{\partial r}\right)\left(\frac{\partial u_r}{\partial z}\right)$$

$$-\left(\frac{\partial u_z}{\partial z}\right)^2 - (u_r)\left(\frac{\partial^2 u_r}{\partial r^2}\right) - (u_z)\left(\frac{\partial^2 u_r}{\partial r \partial z}\right) - (u_r)\left(\frac{\partial^2 u_z}{\partial r \partial z}\right) - (u_z)\left(\frac{\partial^2 u_z}{\partial z^2}\right)$$

$$-\frac{\mu}{4\pi^2\rho r^2}\left\{(I)\left[\left(\frac{\partial^2 I}{\partial r^2}\right) - \frac{1}{r}\left(\frac{\partial I}{\partial r}\right) + \left(\frac{\partial^2 I}{\partial z^2}\right)\right] + \left(\frac{\partial I}{\partial r}\right)^2 + \left(\frac{\partial I}{\partial z}\right)^2\right\}$$

SIMULATION USES ADAPTIVE UNSTRUCTURED GRID

An adaptive unstructured grid is used both to track free surface movement and to resolve small internal features (e.g. boundary layers) The unstructured grid is defined by a list of arbitrarily located (r_j, z_j) points, with flags identifying boundary vs. internal subregion points. The grid is Delauney triangulated via a quick version of Bowyer's algorithm implemented with quadtrees and linked list data structures.

All functions of (r,z) are approximated on this grid using piecewise-linear pyramidal shape functions, $\varphi_j(r,z)$, one for each grid point. Each shape function equals 1 at its own gridpoint, 0 at all other gridpoints, and varies linearly on each triangle connecting gridpoints. An arbitrary function f(r,z) is approximated as $\tilde{F}(r,z) = \sum_j F_j \varphi_j(r,z)$ where the coefficients F_j are selected to make the residual orthogonal in the Galerkin sense to each shape function, i.e. $\langle \varphi_i, (\tilde{F} - f) \rangle = 0$ for every i. This can be rewritten as the vector-matrix equation, AF=b where values of $b_i = \int f(r,z)\varphi_i(r,z) r dr dz$ can be calculated e.g. by numerical integration and stored in the points list, and where nonzero values of $a_{ij} = \int \varphi_i \varphi_j r dr dz$ are evaluated from formulas using triangle vertex coordinates and stored in data structures for triangle edges or for points. F is then calculated iteratively, typically converging to less than 1% error within 10 damped Jacobi iterations.

Galerkin Derivatives and Gradients

A derivative of an interpolated function, e.g. $\frac{\partial F}{\partial z}$, is discontinuous and so cannot itself be differentiated further. However, it can be approximated by a continuous interpolated function, $H(r,z)=\sum_j H_j\varphi_j$ with $\left\langle \varphi_i,\ (\sum_j F_j\frac{\partial \varphi_j}{\partial z}-\sum_j H_j\varphi_j)\right\rangle=0$. This can be written in vector-matrix form as AH=EF for the z-derivative, or AH=CF for the radial derivative. This algorithm and its subsequent relaxation solution is a heavily used part of the simulation code. The nonzero coefficients of A, C, and E are calculated via simple formulas using point coordinates, and stored in the points and edge lists.

The simulation starts each time-step's algorithms by using the stored functions, I, u_r, u_z to calculate $(\frac{\partial I}{\partial r}),(\frac{\partial I}{\partial z}),(\frac{\partial u_r}{\partial r}),(\frac{\partial u_r}{\partial z}),(\frac{\partial u_z}{\partial r}),(\frac{\partial u_z}{\partial z})$, and then use these derivatives to calculate $\left(\frac{\partial^2 u_r}{\partial r^2}\right),\left(\frac{\partial^2 u_r}{\partial z^2}\right),\left(\frac{\partial^2 u_z}{\partial r^2}\right),\left(\frac{\partial^2 u_z}{\partial z^2}\right)\left(\frac{\partial^2 I}{\partial r^2}\right),\left(\frac{\partial^2 I}{\partial z^2}\right)$, all as piecewise-continuous functions. Then linear interpolates are formed to continuously approximate

$$\left(\frac{\partial I}{\partial t}\right) = -(u_r)\left(\frac{\partial I}{\partial r}\right) - (u_z)\left(\frac{\partial I}{\partial z}\right) + \frac{1}{\sigma\mu}\left[\left(\frac{\partial^2 I}{\partial r^2}\right) - \frac{1}{r}\left(\frac{\partial I}{\partial r}\right) + \left(\frac{\partial^2 I}{\partial z^2}\right)\right] + \frac{2}{r}(u_r)(I)$$

which is used with explicit Euler integration to determine the later value of I at each interior point.

A provisional velocity field is formed by explicit Euler integration using

$$\frac{\partial \tilde{u}_r}{\partial t} = -(u_r)\left(\frac{\partial u_r}{\partial r}\right) - (u_z)\left(\frac{\partial u_r}{\partial z}\right) + \nu\left[\left(\frac{\partial^2 u_r}{\partial r^2}\right) + \frac{1}{r}\left(\frac{\partial u_r}{\partial r}\right) - \frac{1}{r^2}(u_r) + \left(\frac{\partial^2 u_r}{\partial z^2}\right)\right]$$

$$- \frac{\mu}{4\pi^2 \rho r^2}(I)\left(\frac{\partial I}{\partial r}\right)$$

$$\frac{\partial \tilde{u}_z}{\partial t} = -(u_r)\left(\frac{\partial u_z}{\partial r}\right) - (u_z)\left(\frac{\partial u_z}{\partial z}\right) + \nu\left[\left(\frac{\partial^2 u_z}{\partial r^2}\right) + \frac{1}{r}\left(\frac{\partial u_z}{\partial r}\right) + \frac{\partial^2 u_z}{\partial z^2}\right] + g_z$$

$$- \frac{\mu}{4\pi^2 \rho r^2}(I)\left(\frac{\partial I}{\partial z}\right)$$

Its divergence is calculated to be the PPE equations' right hand side.

After PPE solution, the pressure field gradient is extracted and used to correct the velocity field at nonconstrained points.

Solution of the PPE by relaxation required excessive iterations (by several orders of magnitude). This has been corrected by incorporating a multigrid scheme to accelerate convergence. (The multigrid scheme is now being debugged.)

Free surface boundary nodes are moved during each integration time step to track free surface motion. The free surface grid velocity is $\vec{v}_b = (\vec{u} \bullet \hat{n})\hat{n}$ Time step duration is restricted so that total boundary motion is smaller than triangular element size.

Internal nodes are reallocated, retired, or repositioned between time steps, and the new grid is retriangulated. Computed point variables are then reinterpolated so that the new field closely matches the old one.

SIMULATION IS Coded in DEC-extended Fortran, residing on PPPL's VAX cluster.

DESIRED: A MORE SIMPLIFIED, APPROXIMATE MODEL

If viscosity is ignored, the Navier-Stokes equations are replaced by the Euler equations which ignore the boundary layer at each liquid-solid surface and thus can use coarser grids. For the steady-state axisymmetric case with only poloidal currents and poloidal motion, the Euler equations are approximated for large r as

$$(\vec{u} \bullet \nabla)\vec{u} = -\nabla\left[\frac{p}{\rho} + gz + \frac{\mu I^2}{8\pi^2 \rho r^2}\right] \quad \text{and} \quad \nabla \bullet \vec{u} = 0 \Rightarrow \nabla \bullet \frac{\partial \vec{u}}{\partial t} = 0$$

with the layer's free and confined surfaces as streamlines.

The toroidal field stream function is the solution of $(\vec{u} \bullet \nabla)I = \frac{1}{\sigma\mu}\Delta^* I + \frac{2u_r I}{r}$,

with the two layer surfaces assuming constant boundary values of I.

A simplified approximate solution of these equations is being sought.

Summary

The Electromagnetic Restraint Concept holds the possibility of many synergistic benefits. However, it is unknown whether it is realistic, due to difficulties in analyzing its MHD behavior. Computer codes capable of evaluating it were not available prior to the APEX effort. A 2-D axisymmetric simulation code without a plasma model has now been coded but has not yet been successfully debugged. To pursue the EMR concept further requires the following:

1. Use of a functioning 2-D axisymmetric free-surface LMMHD computer model to evaluate proposed EMR fields,
2. Marrying the LMMHD model with a plasma model such as TSC to evaluate axisymmetric plasma interactions,
3. Extending LMMHD analysis to evaluate nonaxisymmetric perturbations such as penetrations.

Near-term APEX efforts should focus on concepts such as CliFF for which LMMHD interactions are less severe than in the EMR concept, and which can be analyzed with existing codes. EMR analysis tools should be applied to those concepts and extended further on a low level of effort, as time and resources permit.