Compute Equilibrium and Coil Currents

11/13/05

1) The functionality for the component

In the "evolving equilibrium" description, the momentum equation is replaced by the force balance equation:

$$\rho\vec{V}\bullet\nabla\vec{V}+\nabla\bullet\vec{\vec{P}}=\vec{J}\times\vec{B} \tag{1}$$

where ρ is the plasma mass density, $\vec{V}$ is the fluid velocity, $\vec{\vec{P}}=p\vec{\vec{I}}+\vec{\vec{\Pi}}$ is the plasma pressure tensor, $\vec{B}$ is the magnetic field, and $\vec{J}=\mu_0^{-1}\nabla\times\vec{B}$ is the electrical current density[1]. For an axisymmetric plasma with zero velocity and a scalar pressure, $\vec{\vec{P}}=p\vec{\vec{I}}$, we can take the magnetic field to be of the form:

$$\vec{B}=\nabla\phi\times\nabla\psi+g(\psi)\nabla\phi \tag{2}$$

and (1) is equivalent to the Grad-Shafranov-Schlüter (GSS) equation:

$$R^2\nabla\bullet\frac{1}{R^2}\nabla\psi+R^2\mu_0\frac{d}{d\psi}p(\psi)+\frac{1}{2}\frac{d}{d\psi}g^2(\psi)=0 \tag{3}$$

Here we have introduced the common notation $\Delta^*\psi\equiv R^2\nabla\bullet R^{-2}\nabla\psi$. There are 2 "free functions" that must be specified to solve the GSS equation. In (3), these are the pressure p(ψ) and the toroidal field function g(ψ), but these are not always the most physically meaningful as is discussed below. Also, boundary conditions need to be supplied.

The adiabatic invariants should be exactly the same before and after the equilibrium solution is called. One way to insure this is to write the equilibrium equation in terms of the adiabatic invariants; namely the entropy density $\sigma(\Phi)$ and the transform $\iota(\Phi)$. Using the relations:

$$p=\sigma/V'^{5/3} \tag{4a}$$

$$g=\frac{2\pi}{V'\left\langle 1/R^2\right\rangle} \tag{4b}$$

where the prime denotes derivatives with respect to the toroidal flux Φ, and introducing the new notation for derivatives with respect to the poloidal flux per radian ψ:

$$\dot{V}\equiv\frac{dV}{d\psi}=\frac{2\pi}{\iota}\frac{dV}{d\Phi}=\frac{2\pi}{\iota}V' \tag{5}$$

we can write the GSS equation (3) as follows:

---

[1] $\mu_0=4\pi\times10^{-7}$ henry/m is the permeability of free space

$$\Delta^{*}\psi + R^{2}\frac{d}{d\psi}\left[\left(\frac{2\pi}{i(\psi)\dot{V}}\right)^{5/3}\sigma(\psi)\right] + \frac{(2\pi)^{2}}{i(\psi)\dot{V}\left\langle R^{-2}\right\rangle}\frac{d}{d\psi}\left[\frac{1}{\imath(\psi)\dot{V}\left\langle R^{-2}\right\rangle}\right] = 0 \qquad (5)$$

Special techniques have been developed for solving this in fixed [1,2,5] and free [3,4] boundary codes. Note that solving (5) implies contouring and performing surface averaging during the GSS iteration.

The process of contouring and surface averaging involves defining a flux coordinate system based on the poloidal flux per radian ψ, an arbitrary poloidal angle (the short way around the torus) θ, and the standard toroidal angle (the long way around the torus) ϕ, which has the property that $|\nabla\phi|^{2} = 1/R^{2}$ and that all derivatives with respect to ϕ are zero in an axisymmetric system. We can regard the cylindrical coordinates *(R, Z)* to be functions of *(ψ,θ)*. This is normally called the inverse representation: i.e. solving for *R(ψ,θ)* and *Z(ψ,θ)*. There are two approaches applying the boundary conditions for (5) which we refer to as *fixed boundary* and *free boundary*.

In the fixed boundary description, the outermost ψ=constant surface, is prescribed as a function of time. We call this the limiting surface and denote the flux value there as ψ_{ℓ}. This amount to prescribing $\psi_{\ell}(t)$ as well as $R(\psi_{\ell},\theta,t)$ and $Z(\psi_{\ell},\theta,t)$ for all time *t* during the computation. There are *inverse equilibrium* codes that have been developed to accept these boundary conditions[1,2].

In the free boundary description, the equilibrium equation is applied everywhere with boundary conditions being applied at infinity. This requires the specification or calculation of external coil currents. The shape and value of flux at the limiting flux surface are determined as part of the solution method. The poloidal field coils and metallic structures obey circuits equations coupling them to the plasma:

$$\frac{d}{dt}\left[L_{i}I_{i}(t) + \sum_{i\neq j}M_{ij}I_{j} + \int_{plasma}J(\vec{R}')G(\vec{R}_{i},\vec{R}')d\vec{R}'\right] + R_{i}I_{i} = V_{i}, \quad i = 1, N \qquad (6)$$

Here L_i is the self inductance of circuit or conductor element with current I_i, M_{ij} is the mutual inductance between circuits i and j, the integral represents the mutual coupling with the plasma, R_i is the resistance of this circuit element, and V_i is the applied voltage. In terms of these currents, the poloidal flux function at the boundary of the computational domain can be represented as:

$$\psi_{b}(\vec{R}) = \int_{plasma}J(\vec{R}')G(\vec{R},\vec{R}')d\vec{R}' + \sum_{i=1}^{N}I_{i}G(\vec{R},\vec{R}_{i}) \qquad (7)$$

The Green's function appearing in (6) and (7) is the standard expression for the vector potential due to an axisymmetric current source:

$$G(\mathbf{R};\mathbf{R}') = \frac{\sqrt{RR'}}{2\pi k}\left[(2-k^2)K(k^2) - E(k^2)\right]$$
$$k^2 = \frac{4RR'}{\left[(R+R')^2 + (Z-Z')^2\right]}, \qquad K, E \text{ elliptic integrals} \tag{8}$$

The two-dimensional integrals over the plasma appearing in (6) and (7) are expensive to evaluate, but one can utilize the virtual casing theorem (equivalent to Green's second identity) to convert them to one-dimensional line integrals:

$$\int_{plasma} J(\vec{R}')G(\vec{R},\vec{R}')d\vec{R}' = \oint_{boundary} \frac{d\ell}{R} G(\vec{R},\vec{R}')\frac{\partial U}{\partial n}$$
$$\Delta^* U = \Delta^* \psi, \qquad U = 0 \text{ on boundary} \tag{9}$$

The TEQ code [3] solves equations (5)-(9). The TSC code solves equation (3) to initialize the equilibrium, but actually time advances the equilibrium according to:

$$\rho F\left(\frac{\partial \vec{V}}{\partial t} + \frac{1}{\tau}V\right) + \nabla p = \vec{J} \times \vec{B}$$
$$\frac{\partial \vec{B}}{\partial t} = \nabla \times \left[\vec{V} \times \vec{B} - \eta_{\|}^*(\vec{J} - \vec{J}_{CD})\right], \quad \mu_0 \vec{J} = \nabla \times \vec{B} \tag{10}$$
$$p = \frac{\sigma(\Phi)}{V'^{5/3}}$$

Here $\eta_{\|}^*$ and $\vec{J}_{CD}$ are the modified parallel resistivity and current drive terms as described in the "advance adiabatic variables" component write-up. The parameters *F* and τ appearing in (10) are numerical parameters chosen to speed convergence of the equivalent relaxation iteration for the velocity. The actual velocity V appearing in (10) is not the physical velocity, and the method is accurate only in the limit where F--> 0. Physically, this method corresponds to artificially slowing down the Alfven waves and adding an artificial damping term. A readily derived energy theorem shows that for τ positive, the energy in the solution will decrease monotonically in time while conserving the proper invariants. Mathematically, it corresponds to a dynamic relaxation iteration method for the equilibrium equation, and just performing one relaxation iteration each time step. The boundary conditions can be determined as discussed in equations (6)-(9), although it is noted that an alternate method can be more direct and has some advantages.

By expanding the Green's function in a Taylor series, we can derive the multipolar expansion of the integrals appearing in (6) and in (7):

$$\int_{plasma} J(\vec{R}')G(\vec{R},\vec{R}')d\vec{R}' \simeq G(\vec{R},\vec{R}_0)I_p + \tfrac{1}{2}\mathbf{K}\vdots\nabla\nabla G(\vec{R},\vec{R}_0) \tag{10a}$$

where the monopole and quadrupole functions on the right are evaluated at the current centroid, in order to make the dipole moment vanish. Here, we have defined:

$$I_p = \int_{plasma} J(\vec{R}) d\vec{R} \tag{10b}$$

and

$$\mathbf{K} = \int_{plasma} J(\vec{R})(\vec{R} - \vec{R}_0)(\vec{R} - \vec{R}_0) d\vec{R} \tag{10c}$$

This method has some advantages in that it has a direct physical analogy. It is also very accurate in calculating Volt-Second consumption, and can easily handle open-field line configurations such as occur during disruptions or during Coaxial Helicity Injection (CHI).

Possible extensions to the existing codes would be equilibrium codes that include flow in the equilibrium solution. The easiest would be toroidal flow only, but toroidal and poloidal flow should be considered. Another extension is to include non-scalar pressure. The first step would be to assume the CGL form:

$$\vec{\vec{\mathbf{P}}} = p_\perp \vec{\vec{I}} + (p_\parallel - p_\perp) \frac{\vec{B}\vec{B}}{B^2} \tag{11}$$

A possible extension would be to utilize the full distribution function in calculating the pressure tensor:

$$\vec{\vec{\mathbf{P}}}(\vec{R}) = \int d^3\vec{V} f(\vec{R},\vec{V})(\vec{V} - \vec{V}(\vec{R}))(\vec{V} - \vec{V}(\vec{R})) \tag{12}$$

[1] J. DeLucia, S. Jardin, and A. Todd, "An Iterative Metric Method for solving the Inverse equilibrium Problem", J. Comp. Phys. **37** 2 (1980)

[2] L.E. Zakharov and A. Pletzer, "Theory of perturbed equilibria for solving the Grad-Shafranov equation", Physics of Plasmas **6,** 4693 (1999)

[3] L. L. LoDestro and L. D. Pearlstein, "On the Grad-Shafranov equation as an eigenvalue problem, with implications for q-solvers" Phys. Plasmas **1**, 90 (1994)

[4] S.C. Jardin, N. Pomphrey, and J. DeLucia, "Dynamic Modeling of Transport and Positional Control in Tokamaks", J. Comput. Phys. **66,** 481 (1986)

[5] K. Ling and S. Jardin "The Princeton Spectral Equilibrium Code: PSEC", J. Comp. Phys. **58** 300 (1984)

2) Determine the minimal interface to provide this functionality.

Generic form of the inputs – What informatiion is required by this component.

INPUT REQUIRED BY THIS ROUTINE FROM THE PLASMA STATE:

- profiles of pressure,and transform, $\sigma^n(\Phi)$, $\iota^n(\Phi)$ and possibly rotation $\Omega^n(\Phi)$.

These are real*8 arrays of dimension NPSIT ~ 500, normally on an equally spaced grid in $\Delta\Phi$.

- either old time coil currents and prescription for voltages or new time coil currents (free boundary) I_i . There could typically be 1000 or more currents, but only 10 or so voltage prescriptions.
- PV interface (fixed boundary). This would be $R(\psi_\ell, \theta)$ and Z(ψ_ℓ, θ) These would be two real*8 arrays of about 100 entries.

OTHER INPUT REQUIRED:

- time step Δt
- geometry (vessel, coils, limiters, walls), grid,

The generic form of the outputs – What information will this component provide?

OUTPUT PROVIDED TO THE PLASMA STATE:

- Free boundary: poloidal flux on R,Z grid, (possibly new coil currents) ψ(R,Z): typically a real*8 scalar array of dimension 128 × 128.
- Fixed boundary: cartesian coordinates in inverse representation, $R(\psi,\theta)$ and $Z(\psi,\theta)$. typically on a real*8 scalar grid of dimension 128 × 128.
- metric quantities, and PV interface

OTHER:

special quantities needed for graphics and or diagnostics

What interfaces will be required to access the data this component needs from other components, and what interfaces will it provide to other components

This routine needs to access a component to calculate the surface averages, namely V' and $\left\langle 1/R^2 \right\rangle$. These are repeatedly required during the equilibrium iteration.

3) Identify code specific data

The code needs a complete specification of the geometry and switches and logic to specify what form of equilibrium is required: fixed or free boundary, coil currents or voltages, what type of voltage feed back system to apply.

4) What expectations or assumptions that the component will have about the data it gets from another component.

Not applicable.

5) Identify possible shared infrastructure components such as global time-keeper, interpolators from one representation to another, flux surface average routines, storage of the component's state for restart.

The flux surface average routines needed by this routine are likely needed by other routines.

6) **Identify physics analysis/development needs and identify possible mathematical or algorithmic problems or opportunities for improvement. Identify additional opportunities for parallelism that are not presently realized.**

   The existing equilibrium codes are not parallel. Converting them to parallel MPI codes would probably be worthwhile, but only a modest degree of parallelism could be expected to benefit them. Converting the codes to handle equilibrium velocities and non-scalar pressures would be worthwhile.

7) **Summary of computational needs this component will have when used for the fast MHD campaign. Including computations, memory, amount of time to carry out its designated task in the framework, and kind of parallelism (threaded or distributed memory or both).**

   The memory requirements are very modest.