# SIESTA: A Scalable Iterative Equilibrium Solver for Toroidal Applications

S. P. Hirshman[1], R. Sanchez[1], V. E. Lynch[1], E. F. D'Azevedo, J. C. Hill[1]

[1] *Oak Ridge National Laboratory, Oak Ridge, TN., USA*

Magnetohydrodynamic (MHD) equilibria in toroidally-confined plasmas are computed with various levels of approximation. In the simplest case, a geometrical symmetry of the plasma (such as toroidal symmetry in a tokamak) reduces the problem to a single equation (the Grad-Shafranov equation) for the 2-D magnetic flux function which nevertheless requires numerical solution. In 3-D geometry (rippled tokamaks, stellarators), the assumption of toroidally-nested magnetic flux surfaces is sufficient to allow efficient numerical solution methods based on coordinate inverse methods [1]. However, the presence of magnetic islands can often lower the energy of 3-D configurations and can not be described using inverse methods.

The SIESTA code described here builds upon the works of Chodura and Schlüter [2] and Harafuji [3] to develop an iterative method for minimizing the total (magnetic and plasma) energy in 3D plasmas:

$$W = \oint dV\left(\tfrac{1}{2\mu_0}B^2 + p/(\gamma-1)\right) \qquad \mathbf{(1)}$$

The ideal MHD equations are used to obtain *finite* constrained independent variations of the magnetic field (**B**) and pressure (p), corresponding to discrete versions of Faraday's Law and particle conservation, combined with ideal Ohm's Law ($E + v \times B = 0$) and adiabaticity ($p = n^{\gamma}$):

$$\begin{aligned} \delta B(\xi) &= \nabla\times(\xi\times B) \\ \delta p(\xi) &= (\gamma-1)\xi\bullet\nabla p - \gamma\nabla\bullet(p\xi) \end{aligned} \qquad \mathbf{(2)}$$

The perturbed MHD displacement $\boldsymbol{\xi}=\mathbf{v}\Delta t$ is treated as an independent 3D *variational* parameter and can be used to find a stationary (local) minimum energy state corresponding to the ideal MHD force balance:

$$F_{MHD} \equiv J\times B - \nabla p = 0 \qquad \mathbf{(3)}$$

where $\mu_0 J = \nabla\times B$ is the plasma current.

SIESTA departs significantly in a number of ways from the traditional iterative methods of finding a solution to Eq. (3). As discussed in more detail below, SIESTA solves a set of coupled differential equations for the displacement vectors and does *not* require following magnetic field lines at any point in the computation [3, 4]. This may be significant for accurately resolving magnetic fields in stochastic regions where the randomization of magnetic field lines leads to large spatial integration paths and computation times [5].

We now consider some of the specific features used in SIESTA. First, the 3-D inverse coordinates from the VMEC equilibrium code [1] are used as a background coordinate system [6] for the equilibrium computations. While the introduction of these non-orthogonal curvilinear coordinates complicates the finite difference equations, it has the appealing physical advantage of providing a nearby equilibrium state from which the displacements in Eq. (2) are expected to be small, even when the initial closed magnetic surface topology is broken by islands.

The second distinction of SIESTA is that it uses a Newton solver (based on the *expected* boundedness of the displacements) to rapidly converge to equilibrium (in typically 10's of iterations). This contrasts to using a series of very small displacements (corresponding to a small time step in the momentum equation, not used here) to iterate Eq. (2) many (1000+) times to reach the equilibrium state in Eq. (3). The Newton iteration used in SIESTA is obtained from the field and pressure perturbations given by Eq. (2), used in Eq. (3) to update the variational displacements. With $B_{n+1} = B_n + \delta B(\xi), p_{n+1} = p_n + \delta p(\xi)$ :

$$F_{n+1}(\xi) \equiv F_n + \delta J(\xi) \times B_n + J_n \times \delta B(\xi) - \nabla \delta p(\xi) + O(\xi^2) = 0 \qquad \textbf{(4)}$$

Here, $F_n$ ($B_n$ ,$J_n$) is the MHD force (magnetic field, current) at the n-th (previous) iteration step, and the second order term in ξ is $\delta J \times \delta B$. If this term is ignored, then Eq. (4) can be converted to a linear matrix equation of the form:

$$H_{ij} x_j = f_j \qquad \textbf{(5)}$$

Here, the $x_i$ are comprised of the radially discretized Fourier harmonics of the contravariant components $\sqrt{g}\,\xi \bullet \nabla \alpha$, where $\alpha \in (s,u,v)$ represents the VMEC curvilinear flux (in SIESTA, $s = \sqrt{\Phi}$ is the "polar" radius) and angle coordinates (u,v).

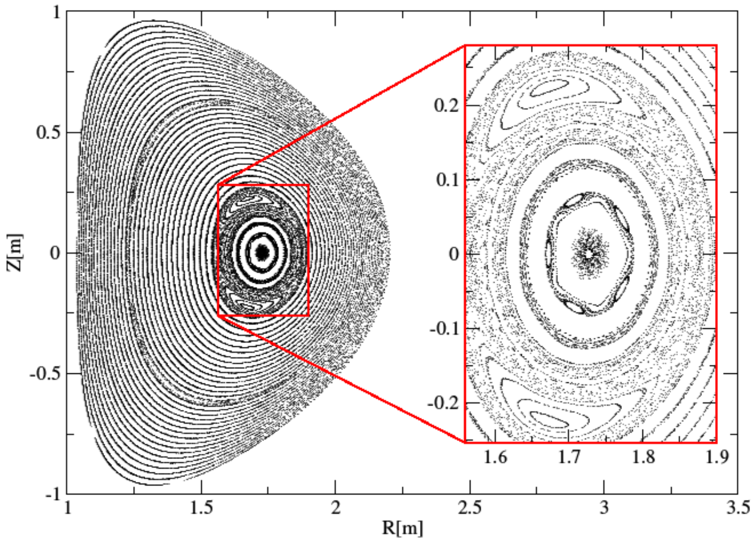
A third unique feature of SIESTA is based on the observation that the highest-order differential operators in Eq. (4) are second-order in the radial coordinate. This implies that the Hessian matrix in Eq. (5) is *block tri-diagonal.* It is possible to solve Eq. (5) *exactly* using a scalable and efficient algorithm. The algorithm avoids fill-in so that efficient storage is maintained during the solution of Eq. (5). This is important for future applications of SIESTA to analyze high temperature, ITER-like plasma conditions which will require very fine radial resolutions to resolve islands at a large magnetic Reynolds number *S*.

Note that *all* relevant space scales of linearized ideal MHD – compressional and shear Alvén, and sound wave scales – are included in the Hessian matrix **H** in Eq. (5). Therefore, the condition number of **H** can be quite large, especially at fine radial resolutions. The solution of Eq. (5) therefore requires an appropriate preconditioner to avoid numerical inaccuracies. Let **P** be a matrix that is invertible and for which $\mathbf{P}^{-1}\mathbf{H} \sim \mathbf{I}$. Such a preconditioner can be obtained in a number of ways, by adding (1) a small Levenberg-Marquardt diagonal element to **H** or (2) a small parallel velocity damping term $\sim \mu_{\|}\xi_{\|}$ to the linearized force, which eliminates the approximate null space of **H** for small wavelength displacements parallel to **B**. This latter method works well in general and has been used in the calculations described here. Note that the MHD force in Eq. (4) is not modified by $\mu_{\|}$ since *only* the preconditioner includes this term for numerical stability.

SIESTA uses the reverse-communication GMRES package available from CERFACS [7] to solve Eq. (5) iteratively. Right-preconditioning provides the best convergence of the GMRES algorithm for our problem. Note that the importance of the preconditioner is that it coalesces the disparate Alfvènic and sonic scales thus making the condition number close to unity. Typically 100 iterations of GMRES are sufficient to solve the linearized forces to a normalized squared force residual ~ $10^{-20}$. A nonlinear iteration loop - *n* in Eq.(4) – is required to solve the MHD force Eq. (3) and requires tens of iterations to reduce the nonlinear force residual from an initial value of ~$10^{-5}$ (the interpolated nested VMEC solution) to ~ $10^{-20}$.

When running SIESTA, a simple tri-diagonal solver is first used to reduce the initial force residuals arising from interpolating the VMEC equilibrium onto the SIESTA mesh. A set of resonant perturbations is applied at low order rational surfaces to break the surfaces. In addition, operator splitting is used to apply a resistive perturbation $\delta B_{n+1} \sim -\nabla \times \eta J_n$ to the magnetic field perturbation for low iteration values, $n \leq 5$, in Eq. (4). This diffuses resonant current sheets, allowing islands to expand to decrease the magnetic energy in Eq. (1).

An equilibrium solution obtained by SIESTA for a D3D-like tokamak case with <β>~2.9% is shown in Fig. 1 for the v=0 toroidal plane. The safety factor profile (Fig. 2) has two widely separated q=2 resonances at normalized minor radii s/a~0.25 and 0.70. The blow-up shows the details of the innermost resonant islands. The q=5/2 island chain is also clearly visible. The energy change compared with the initial nested surface equilibrium is $\delta W/W$~-$10^{-5}$. The number of points used for this calculation were $N_s$=101 (radial nodes), $M_u$=13 (poloidal modes), $M_v$=7 (toroidal modes), corresponding to 27573 independent variables and equations. The amount of CPU-time required was less than five minutes on a single Pentium processor desktop computer. Present day simulations for $S \sim 10^4$ run with $\sim 10^2$ radial points, whereas a full volume ITER simulation with $S \sim 5\times10^8$ will require $\sim 10^4$ radial points. Angular resolution will also increase with an expected dense block structure approaching $(10^3\text{-}10^4)^2$ elements. Scalable solvers are being investigated that distribute the blocks on parallel processors. High resolution ITER calculations with SIESTA will require petaflop computing on massively parallel resources.



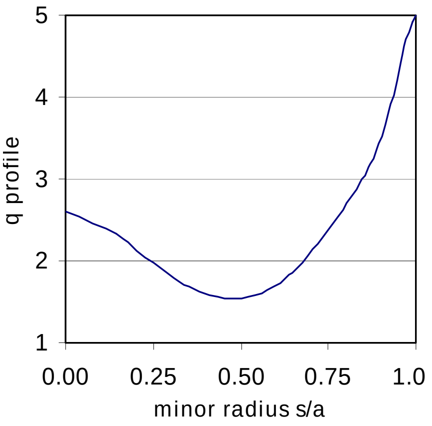


**Fig 1 Tokamak equilibrium with islands at the q=2 resonant surfaces (m=2, n=-1)**

**Fig 2 Safety factor q**

## References

[1] S. P. Hirshman, W. I. van Rij, and P. Merkel, Comput. Phys. Commun. **43**, 143 (1986).

[2] R. Chodura and A. Schlüter, J. Comput. Phys. **41** (1981) 68.

[3] K. Harafuji, T. Hayashi, T. Sato, J. Comput. Physics **81**, 169 (1989).

[4] A. H. Reiman and H. S. Greenside, Compt. Phys. Commun. **43**, 157 (1986).

[5] S.R.Hudson, Physics of Plasmas **11**(2) 677 (2004).

[6] T. C. Hender, B. A. Carreras, L. Garcia, J. A. Rome, and V. E. Lynch, J. Comput. Phys. **60**, 76 (1985).

[7] V. Frayssé, L. Giraud, S. Gratton, and J. Langou, CERFACS Technical Report TR/PA/03/3, (http://www.cerfacs.fr/algor/Softs/GMRES/index.html) 2003.