THE CURRENT STATUS AND FUTURE OF THE SIESTA EQUILIBRIUM CODE

by

Carson R Cook

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science (Electrical and Computer Engineering)

at the UNIVERSITY OF WISCONSIN-MADISON

2011
The SIESTA magnetohydrodynamic equilibrium code is used to find a stable plasma equilibrium in toroidal devices such as stellarators and tokamaks. Previous equilibrium codes, such as VMEC, did not allow magnetic surfaces to tear and thus only resolved configurations with closed, nested flux surfaces. SIESTA uses a static, background coordinate system and includes resistivity, allowing current sheets to open into magnetic islands. Through a nonlinear energy minimization process, SIESTA finds a stable, lower-energy equilibrium if the initial starting equilibrium is unstable to perturbations applied at rational surfaces. The code is introduced, followed by a discussion of the research done on and problems found with a pentadiagonal solver. An initial study of a nullspace inherent to the problem is presented next, followed by some post-processing results studying the structure of magnetic islands.
Contents

1 Introduction 6

2 Newton’s method for energy minimization 9

3 Implementation and study of the pentadiagonal linear solver 11

4 Characterization of the nullspace 18

5 Post-processing using the VisIt visualization tool 33

6 Conclusions 43

A HSX hill and well equilibria 45
List of Figures

4.1 $\cos \alpha$ for eigenvector with eigenvalue of $-1.43985928512053334 \times 10^{-8}$. 22
4.2 $\cos \alpha$ for eigenvector with eigenvalue of $-6.07887922906006218 \times 10^{-8}$. 24
4.3 $\cos \alpha$ for eigenvector with eigenvalue of $-2.37770352903102343 \times 10^{-7}$. 26
4.4 $\cos \alpha$ for eigenvector with eigenvalue of $-2.09812505620376597 \times 10^{-7} +$
   $i1.01866740563547538 \times 10^{-8}$. 28
4.5 $\cos \alpha$ for eigenvector with eigenvalue of $-2.09812505620376597 \times 10^{-7} -$
   $i1.01866740563547538 \times 10^{-8}$. 30
4.6 $\cos \alpha$ for eigenvector with eigenvalue of $-5.1055121286084679$. 32

5.1 $|B|$ on last-closed-flux-surface of stellarator under study. 34
5.2 Iota profile displaying 6/1 and 5/1 resonance locations (taken from [1]). 35
5.3 Poincare plot displaying the 6/1 island chain and smaller 5/1 chain
   (taken from [1]). 36
5.4 Single 6/1 island structure. 38
5.5 Combined 6/1 island chain. 39
5.6 7/1 island chain. 40
5.7 Combined 6/1 and 7/1 island chains. 41

A.1 Iota profile for 6.2% hill in HSX. 46
A.2 Pressure contours for 6.2% hill in HSX using 20 toroidal modes. 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.3</td>
<td>Pressure contours for 6.2% hill in HSX using 10 toroidal modes</td>
<td>48</td>
</tr>
<tr>
<td>A.4</td>
<td>Iota profile for 6.2% well in HSX</td>
<td>49</td>
</tr>
<tr>
<td>A.5</td>
<td>Pressure contours for 6.2% well in HSX using 20 toroidal modes</td>
<td>50</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Ideal magnetohydrodynamic equilibrium codes such as VMEC [2] assume closed, nested magnetic flux surfaces. The SIESTA (Scalable Iterative Equilibrium Solver for Toroidal Applications) code can resolve equilibria with topology changes allowing for magnetic islands and stochasticity [1]. The code begins with a VMEC equilibrium with nested surfaces. It then perturbs the equilibrium at rational surfaces. Current sheets will form at these rational surfaces in ideal MHD, so finite resistivity is included to diffuse away the current sheets. This diffusion allow the islands to open up if the original equilibrium is unstable and there is a lower energy state available. Newton’s method is used as the nonlinear energy minimization solver to find this lower energy state containing islands. At each iteration of Newton’s method, a linearized problem must be solved. This new equilibrium that includes islands can be used to initialize other extended MHD simulations or reconstruction codes. SIESTA could also be used as the equilibrium engine in optimization codes such as STELLOPT that study new confinement concepts. In this way, configurations can be developed that are optimized against island formation.

A proper solution method must be found for numerically solving the linear system
obtained at each iteration of Newton’s method. The Hessian matrix computed in SIESTA has a block tridiagonal structure. However, due to the presence of both stable and unstable modes in the magnetohydrodynamics system under study, the Hessian matrix is not in general positive definite. Since Newton’s method requires a positive definite matrix for a decrease in energy to occur, the unaltered Hessian system will not necessarily lead to convergence to an equilibrium with a lower energy. The question thus becomes: is it possible to modify the system such that a decrease in energy can be obtained using Newton’s method, and if so how can this be done effectively without too much computation time and memory usage?

The work flow of running a SIESTA simulation proceeds as follows. Initially a VMEC simulation is ran until convergence. The output of this step will be an MHD equilibrium with closed, nested flux surfaces. The data for this equilibrium is stored in a .wout file. This serves as the input for the SIESTA simulation. The VMEC .wout file is specified in the SIESTA, located in siesta.jcf. Within the input file, other parameters are specified, including the number of poloidal and toroidal modes to be included in the calculation, the number of radial surfaces included, and the number of iterations to be performed. A force tolerance is also specified that determines when the simulation will finish. Other parameters that can be set include the resistivity $\eta$, the location of and strength of the plasma perturbation, and other post-processing and numerical controls. Another parameter that is specified is known as the Levenberg-Marquardt parameter and is used to make the Hessian matrix positive definite. This number currently has to be set by the user of the code, and it controls the convergence of the nonlinear energy minimization process. A “good” choice leads to a simulation that converges quite quickly. A poor choice can give rise to a simulation that may not converge to a lower energy state at all, even if one exists. The problem is that the required LM parameter
varies greatly from configuration to configuration, and there is no effective method for picking what this number should be for a new device without a lot of trial and error. Since ideally there would be no parameters that have to be adjusted manually by the physicists and engineers using the code, this is a problem. In order to try to alleviate this challenge, a pentadiagonal solver has been proposed and tested as described below. While this was shown to be ineffective for reasons discussed in this paper, there are other methods involving nullspace removal that can be investigated in the future to remove the need for manual adjustment.
Chapter 2

Newton’s method for energy minimization

SIESTA solves for the minimum of the nonlinear magnetohydrodynamic energy functional given by

\[ W = \int \left[ \frac{B^2}{2\mu_0} + \frac{p}{\gamma - 1} \right] \, dVol \]  \hspace{1cm} (2.1)

The minimum can be found through a variational energy principle for MHD stability [3],[4]. The configuration of minimum energy will satisfy the ideal MHD force balance equation,

\[ F = J \times B - \nabla p = 0, \]  \hspace{1cm} (2.2)

resulting in an equilibrium. The perturbed force away from equilibrium (which we are always approximately near numerically) is given by

\[ \delta F = \frac{1}{\mu_0} \left[ (\nabla \times B_1) \times B + (\nabla \times B) \times B_1 + (\nabla \times B_1) \times B \right] - \nabla p_1 \]  \hspace{1cm} (2.3)

\[ B_1 = \nabla \times (\xi \times B) \]  \hspace{1cm} (2.4)

\[ p_1 = (\gamma - 1) \xi \cdot \nabla p - \gamma \nabla \cdot (p \xi) \]  \hspace{1cm} (2.5)
Here the subscript 1’s refer to quantities that are first order in $\xi$. Since this perturbed force equation is quadratic (second-order) in the plasma displacement $\xi$ due to the third term, an iterative approach is necessary to solve this energy minimization problem, since no exact solution process exists.

In order to solve this nonlinear optimization problem, Newton’s method is implemented [5]. In this method, the nonlinear energy $W$ is approximated by a quadratic function $\hat{W}$ at each iteration. The quadratic form is obtained from the truncated Taylor expansion of the energy around $\xi = 0$:

$$W \approx \hat{W} = W|_{\xi=0} + \frac{dW}{d\xi}|_{\xi=0} \cdot \xi + \frac{1}{2} \xi \cdot \frac{d^2W}{d\xi^2}|_{\xi=0} \cdot \xi$$  \hspace{1cm} (2.6)

The second derivative of the energy with respect to the displacement is simply the Hessian matrix of the energy functional. For simplicity, this will now be referred to as $H$. In order to minimize the quadratic energy $\hat{W}$ its derivative must be set to zero and the minimizing displacement solved for:

$$\frac{d\hat{W}}{d\xi} = \frac{dW}{d\xi}|_{\xi=0} + \frac{d^2W}{d\xi^2}|_{\xi=0} = 0$$  \hspace{1cm} (2.7)

Rewriting 2.7 using $H$ and $\frac{dW}{d\xi}|_{\xi=0} = -F_{\text{prev}}$ gives the following linear equation that must be solved at each iteration of Newton’s method:

$$H\xi = F_{\text{prev}}$$  \hspace{1cm} (2.8)

This equation is used to find the new displacement $\xi$ for the next iteration by inverting the MHD Hessian matrix and applying it to the force residual from the previous iteration. This is done repeatedly until the force falls below a specified tolerance. The solution of 2.8 will be the topic of the next chapter.
Chapter 3

Implementation and study of the pentadiagonal linear solver

In order to find a stable minimum of the energy functional of the MHD system, Newton’s method is used to linearize the problem at each iteration as described above. Ultimately we would like the nonlinear force to be zero, $F_{\text{nonlinear}} = 0$, signifying an equilibrium. To approach this solution, the linearized force is set to zero at each iteration of Newton’s method. The linear problem resulting from minimizing the quadratic energy is as follows:

$$\mathbf{H}\xi = F_{\text{prev}} \quad (3.1)$$

$$F_{\text{linear}} = \mathbf{H}\xi - F_{\text{prev}} \quad (3.2)$$

Here $F_{\text{linear}}$ must be minimized, corresponding to a numerical solution to the linear equation. $F_{\text{prev}}$ is the full nonlinear force evaluated using the plasma displacement from the previous iteration. Once a new displacement, $\xi$ is found that minimizes the new linear force, it is inserted into the full nonlinear force equation to obtain $F_{\text{prev}}$ for the next iteration. In this equation, the structure of the Hessian matrix $\mathbf{H}$ is block
tridiagonal. This is because the Hessian embodies the finite-differencing of a second-order differential equation in terms of the plasma displacement. This couples every three flux surfaces together: the one being evaluated and its two closest neighbors. Within each block, the spectral representations of the toroidal and poloidal dimensions manifest themselves. Assuming a high mode content to the equilibrium, the blocks will be densely populated. There is a parallel cyclic solver code known as BCYCLIC that has been developed for fast, efficient solution of this block tridiagonal system [6].

When using Newton’s method, descent towards the equilibrium is only guaranteed if the Hessian matrix is positive definite. Since in general the configuration at a given iteration will not be in actual equilibrium, this Hessian will not be positive definite. There will be positive and negative eigenvalues corresponding to the stable and unstable MHD modes of the Alfven eigenspectrum. Only very near the stable equilibrium we are seeking will the eigenvalues become strictly positive, signifying a positive definite matrix.

Due to the general indefiniteness of the Hessian, some modification is necessary in order to get positive definiteness and ensure that Newton’s method will converge. The most straightforward method of modifying the Hessian to become positive definite is by adding a positive ”Levenberg-Marquardt” parameter times the identity matrix to the Hessian as follows:

\[ H \rightarrow H + \mu I \]  \hspace{1cm} (3.3)

However, the problem with this approach is that it is unclear how to properly choose the \( \mu \) parameter. \( \mu \) must be chosen larger than the negative eigenvalue of largest modulus so that the eigenspectrum is shifted completely to the positive side. However, solving
the full eigenvalue problem of the huge SIESTA Hessian matrix

\[ \mathbf{H} \mathbf{x}_n = \lambda_n \mathbf{x}_n \]  \hspace{1cm} (3.4)

is very computationally expensive and usually takes much longer to compute than the equilibrium calculation itself. There are some potentially useful partial eigenvalue solvers (e.g. ARPACK, PROPACK) that could be used to compute just the largest (modulus) negative eigenvalues so that an intelligent choice of \( \mu \) could be made. However, further studies will need to be done to determine the efficiency of these algorithms. There may also be potential improvements possible using Gerschgorin’s Theorem to estimate the eigenspectrum, which I will not detail in this paper.

In lieu of using these algorithms, the technique currently implemented in SIESTA requires manually choosing this parameter to achieve good convergence. This is obviously a much less than desirable situation, since SIESTA is to be run on many different configurations (experimental machines), each of which have different equilibrium/stability parameters leading to a different eigenspectrum and ultimately a different optimal Levenberg-Marquardt parameter. The physicists using the code should not have to manually tweak a parameter to get the code to find the stable equilibrium; it should work correctly after fully specifying their physical configuration and nothing else.

The problems inherent to the above tridiagonal method with LM parameter led to the suggestion of trying to modify the Hessian into a pentadiagonal system which can be made positive definite without a user-chosen parameter. When 3.1 is converted to a pentadiagonal system by squaring the Hessian matrix and modifying the right-hand-side, we obtain

\[ \mathbf{H}^T \mathbf{H} \xi = \mathbf{H}^T \mathbf{F}_{\text{prev}} \]  \hspace{1cm} (3.5)
Note that this equation is analytically identical to the previous one, since we have simply multiplied both sides by $H^T$. Since nothing has been changed in theory, 3.2 can still be used for the force residual. The idea here is that the pentadiagonal Hessian is now positive semi-definite. This is because the left-hand-side of the equation involves the "square" of the original Hessian ($H^T H$) which is the analog of the square of a scalar, and thus the eigenvalues are all non-negative. The only aspect that still must be taken into account is the potential for zero (or tiny) eigenvalues. By adding an offset Levenberg-Marquardt parameter $\mu$ (as before) to the pentadiagonal matrix the result should be a strictly positive definite matrix, giving the desired result. Since the pentadiagonal matrix is positive semi-definite, $\mu$ can be chosen as small as possible, and the result should still be positive definite.

The problem that rears its ugly head in practice, however, is that the condition number becomes the dominant numerical limitation, resulting in a terrible solution of the linear problem. This is the main reason that the pentadiagonal solving technique will not work: the condition number kills the solution. If the condition number of $H^T H$, defined below, is too high then the linear force residual may be much larger than desired, signifying a poor solution to the linear problem at each iteration:

$$\kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$$

(3.6)

As can be seen, the condition number $\kappa$ is the ratio of the largest to smallest eigenvalues. Since we effectively squared the original tridiagonal Hessian in obtaining the pentadiagonal system, we also squared the condition number, making it enormous. A large condition number means a bad solution of a linear equation, so the Newton step obtained will not be a good estimate. Recently, a pentadiagonal case was run with the Levenberg-Marquardt parameter set to zero. In this case I was able to check the
solution using a residual equivalent to the original linear residual. This gave the result

\[ F_{\text{linear}} \cdot F_{\text{linear}} \approx 10^{-10} \quad (3.7) \]

This huge linear residual demonstrates that we have not solved the problem numerically to an acceptable degree of accuracy. Thus the pentadiagonal approach will not work if a zero (or small) LM parameter is chosen because the condition number limits the accuracy of the solution to the problem.

But what happens when a sufficiently non-zero LM parameter is added so that the condition number is more tractable and allows for an acceptable numerical solution to the LM-modified problem? Adding an LM parameter essentially shifts all of the eigenvalues by the same amount, bringing the smallest eigenvalues away from zero and decreasing the condition number. There are three force residuals that must be brought to zero during the course of the problem. Overall, the nonlinear force must be minimized. In order to do this with Newton’s method, the linear force residual given by 3.2 at each iteration must be minimized. Finally, and this is a subtle point, the linear force residual of the LM-modified linear equation resulting from changing \( H \) must be minimized. This modified residual is the only one of these three force residuals that directly corresponds to how well a linear problem is actually solved, and hence depends on the condition number of the problem. As discussed above, a small LM parameter will not suffice, since the condition number dominates. So the modified linear force residual is that of a substantially modified linear equation, given below:

\[ (H^T H + \mu I) \xi = H^T F_{\text{prev}}. \quad (3.8) \]

The original system has now been modified in this formulation, since \( \mu \) is only added to
the left-hand-side of the equation. In addition, it was noted above that this LM parameter is required to be somewhat sizable. As a result, the quality of the solution to this LM-modified equation does not imply an acceptable solution to the actual, unmodified linear force residual that needs to be minimized. That original residual does not include \( \mu \) anywhere. In effect, the original problem to be solved has been greatly changed.

In order to look directly at the pentadiagonal solver’s effectiveness in solving the modified linear problem, a backsolver was implemented by hand. If the above modified pentadiagonal system is solved effectively we would expect the modified force residual given by

\[
F_{\text{mod}} = (H^T H + \mu I) \xi - H^T F_{\text{prev}}
\]  

(3.9)

to go to zero. It is only through this residual formulation that we can analyze whether the pentadiagonal solver is being implemented correctly insofar as properly factoring and solving the modified matrix system is concerned. Upon modifying the linear problem with many different sizable LM parameters, all large enough to eliminate the condition number problem discussed above, the modified linear force residual looked promising, with some runs performing as well as

\[
F_{\text{mod}} \cdot F_{\text{mod}} \approx 10^{-36}.
\]

(3.10)

This tiny modified force residual demonstrated that the pentadiagonal solver was indeed doing its job of solving the modified linear system correctly. But the most important question is how well it solved the original, non-LM modified linear system. The original linear force residual did not fare nearly as well, generally performing only slightly better than the unmodified, condition number dominated approach detailed above. Some of the best attempts gave residuals of

\[
F_{\text{linear}} \cdot F_{\text{linear}} \approx 10^{-12}.
\]

(3.11)
This is still a much higher linear force residual than what is needed for any sort of convergence with Newton’s method. These unmodified linear force residuals would need to be no larger than $10^{-15}$ and ideally much smaller than that.

The pentadiagonal solution method simply does not work. With a tiny (or zero) LM parameter, the condition number is squared compared to the tridiagonal case, and a very poor numerical solution of the linear equation is obtained. When a larger LM parameter is added to quell the condition number, the original linear problem becomes so greatly modified that a poor solution to the original linear problem is obtained even when the modified problem is solved to exact precision. Since the pentadiagonal solution was an ineffective way to remove the necessity of a manually adjusted parameter in SIESTA, studies of the nullspace are now underway and detailed in the next section. Suppression of the nullspace may be another potential way to alleviate the situation.
Chapter 4

Characterization of the nullspace

The nullspace of a matrix (or linear operator) \( A \) is defined as the set of vectors satisfying the following equation:

\[
\{ x | Ax = 0 \} \tag{4.1}
\]

Thus the nullspace is the set of eigenvectors with eigenvalues of zero. It is of great interest to understand the nullspace of the SIESTA Hessian matrix for reasons that will be described shortly. When analyzing nullspaces in a numerical computation, however, the above definition does not quite suffice. Due to limits in machine precision, there may be some eigenvectors that fall into an "approximate nullspace" defined by the corresponding eigenvalues being less than some small tolerance.

\[
\{ x | Ax < N_{tol} \} \tag{4.2}
\]

This approximate nullspace behaves as an actual nullspace for the numerical calculation, and the terms approximate nullspace and nullspace will be used interchangably for this problem from here on.

The reason the nullspace causes problems is two-fold. From a theoretical perspec-
tive, if a true nullspace exists (with zero eigenvalues) then the matrix is strictly not invertible. This means that the critical step in each SIESTA iteration in which the Hessian matrix is inverted will not work, resulting in an incorrect new plasma displacement. However, the actual case that will happen in practice is slightly different. As mentioned above, the nullspace that will be obtained in this computer simulation will actually be an approximate nullspace in which the eigenvectors have extremely tiny eigenvalues. If this type of nullspace is present, we get another type of serious problem. Let’s assume that in the linear problem to be solved at each iteration, ie

\[ \mathbf{H}\xi = \mathbf{F}_{\text{prev}}, \]  

(4.3)

the force residual vector on the right-hand-side can be written as a series in terms of the \( n \) eigenvectors of the Hessian matrix as follows:

\[ \mathbf{F}_{\text{prev}} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i. \]  

(4.4)

Now if the coefficients of the eigenvectors that are part of the nullspace are non-zero in general, and in practice large enough, then the next displacement obtained \( \xi \), will be completely dominated by the component of the residual force in the null vector direction. For example, if there is only one non-zero coefficient of a nullspace vector, say vector \( m \), then the new displacement will be almost entirely determined by

\[ \xi = \mathbf{H}^{-1}\alpha_m \mathbf{x}_m. \]  

(4.5)

Since the eigenvector of the nullspace, \( \mathbf{x}_m \), has a tiny eigenvalue, the eigenvalue obtained in the above equation containing the inverse will be enormous, thus leading to a huge, non-physical step in that direction. These large steps can lead to displacements to areas of the domain that will never allow convergence to a stable equilibrium.
Because of these concerns, it would be very nice to study the structure of the nullspace eigenvectors so that they could be potentially removed from the domain of the force residuals, thus removing the above problems. Each eigenvector is only an eigenvector in the sense of the numerical solving of the linear problem at each iteration. From a physics perspective, it is an vector field eigenfunction, consisting of a vector at each point in the computational domain (ie. within the plasma in the confinement device). In order to study the nullspace, a configuration with a small problem size is needed since a full eigenvalue/eigenvector problem must be solved, a task that is very computationally intensive. Using LAPACK’s eigensolvers, the eigenvectors corresponding to the smallest eigenvalues of the Hessian for a classical, low-beta, three field-period stellarator were obtained and studied. These results are very preliminary, and much more research needs to be done. Currently, all of the results are from the initial, unperturbed equilibrium state corresponding to the output of the VMEC code. Further nullspace studies will be done on the equilibrium as it evolves throughout the simulation, converging to the final solution including magnetic islands. The primary interest is the eigenvector’s orientation with respect to the magnetic field. The angle $\alpha$ between the eigenvector and the magnetic field is plotted below as $\cos \alpha$ and thus varies between -1 and 1. The results are shown in three dimensions on the last closed flux surface and in two dimensions on the poloidal plane at $\phi = 0$ for eigenvectors in the null space of several different eigenvalues as well as for an eigenvector with a larger eigenvalue.
Figure 4.1: \( \cos \alpha \) for eigenvector with eigenvalue of \(-1.4398592851205333 \times 10^{-8}\)
Figure 4.2: \( \cos \alpha \) for eigenvector with eigenvalue of \(-6.0788722906006218 \times 10^{-8}\)
Figure 4.3: $\cos \alpha$ for eigenvector with eigenvalue of $-2.3770352903102343 \times 10^{-7}$
Figure 4.4: $\cos \alpha$ for eigenvector with eigenvalue of $-2.09812505620376597 \times 10^{-7} + i 1.01866740563547538 \times 10^{-8}$
Figure 4.5: \( \cos \alpha \) for eigenvector with eigenvalue of \(-2.09812505620376597 \times 10^{-7} - i1.018674056354758 \times 10^{-8}\)
The most apparent conclusions that can be drawn from these plots is that the nullspace eigenvectors are largely parallel (or anti-parallel) to the magnetic field vector. This can be seen by the red and blue dominated plots corresponding to a \( \cos \alpha \) of 1 or -1. The bottom plot displays an eigenvector with a much larger eigenvalue; this eigenvector is not part of the nullspace. It can be seen that the vector is not parallel to the magnetic field in as many positions in the domain. It is still largely parallel, but not as dominated by the parallel as the eigenvectors from the nullspace.
Chapter 5

Post-processing using the VisIt visualization tool

Field-line following is a common tool in plasma physics, and two-dimensional Poincare puncture plots are often created from the results. Usually the plots are created with R, Z axes, and it can be difficult to determine the magnetic topology of a configuration simply by looking at the plot. The VisIt visualization software package developed at Lawrence Livermore [7] includes a Poincare utility that allows multiple puncture surfaces to be displayed [8]. In addition, the software has options to isolate island chains, plot entire field-line following generated surfaces, and connect disjoint puncture points into a coherent magnetic topology. These options can allow islands to be visualized in new ways and is very helpful when trying to see island structures. This is useful when analyzing equilibria obtained from SIESTA.

Due to the machine’s strong helical shaping, HSX is not a good test candidate for the SIESTA code at this time. The high toroidal mode content inherent in the design makes it a difficult machine to analyze efficiently. Instead a classical, low-beta stellara-
tor with no net current was used to demonstrate the new Poincare techniques. This is the same configuration discussed above in the section on the nullspace. The stellarator has three field periods, and its magnetic structure can be seen in 5.1.

Figure 5.1: \(|B|\) on last-closed-flux-surface of stellarator under study

The iota profile for this device is shown below. Note that the convention used here
is

\[ \iota = \frac{n}{m} \quad (5.1) \]
\[ q = \frac{N_{fp}}{\iota} = \frac{3}{\iota} \quad (5.2) \]

where \(N_{fp}\) is the number of field periods. This will be used when discussing the structure of the islands.

Figure 5.2: Iota profile displaying 6/1 and 5/1 resonance locations (taken from [1])

The SIESTA code is now used to perturb the plasma at the rational surfaces shown above. The resulting equilibria will be discussed for the rest of the section. A typical Poincare puncture plot resulting from the output of the simulation is displayed below in 5.3, and a large 6/1 \((m = 6, n = 1)\) island chain is displayed interior to a smaller 5/1 chain.
Figure 5.3: Poincare plot displaying the $\frac{6}{1}$ island chain and smaller $\frac{5}{4}$ chain (taken from [1])
The large 6/1 island chain is easily seen in this plot. However, since there are only dots in the plot corresponding to puncture points, it is much more difficult to see the 5/1 island chain at a larger radius. The high number of points required to resolve the islands also obscures them from view, since there are so many points corresponding to normal closed surfaces. As a result, much of the plot looks black.

It would be advantageous to have a way of isolating the islands and seeing them in three dimensions. This way the connectivity of the islands can be discerned. Using the Poincare option in the VisIt software, this is possible. First, the components of the magnetic field vector are dumped to a SILO data file (.silo) at the conclusion of the SIESTA simulation. Then, the SILO file is read in by VisIt and the B field components are combined into a vector field at every point in the mesh. The Poincare plotter option is used on this resulting field. Field lines in the 6/1 island chain should wrap around twice toroidally for every poloidal transit (since this is a three field period device). Plots 5.4 and 5.5 below demonstrate this behavior, and also show that the total 6/1 chain is actually composed of three distinct island structures, with 5.5 showing the geometry of these islands when all three structures are combined into a single plot. 5.6 shows a single 7/1 island chain internal to the 6/1 chain, and 5.7 shows all of the chains combined. In order to obtain the plots the line option in the Poincare plotter was used with a point density of 4 and 500 to 1000 punctures.
Figure 5.4: Single 6/1 island structure
Figure 5.5: Combined 6/1 island chain
Figure 5.6: 7/1 island chain
Figure 5.7: Combined 6/1 and 7/1 island chains

There is much further work to be done with this, and it would be interesting to see if the $m = 7$ island chain can be resolved and cross-checked either with a traditional Poincare plotter or by looking at the pressure profiles. It appears to be physical, but a more in-depth analysis needs to be done before any conclusive statements can be made. However, this demonstrates the utility of the post-processing and visualization
software included in VisIt (particularly the Poincare plotter) in helping to find specific island chains and magnetic topology features. This will prove useful when analyzing the data that comes out of future SIESTA studies on machines such as HSX, once the SIESTA software can handle these devices. It should also be noted that the Poincare plotter used in VisIt is very fast, finishing all of the plots shown here in under a minute (usually well under a minute).
Chapter 6

Conclusions

As can be seen, there is still a lot of work to be done. However there has been a sizable amount of progress made so far. SIESTA is clearly a very useful tool for studying the presence of islands in three-dimensional MHD equilibria. The proposed pentadiagonal solver proved to be more trouble than it was worth for this application, but other techniques are actively being pursued, including removal of the nullspace. Analysis of the nullspace is underway but more work must be done in characterizing it. Once that is accomplished, the next task will be determining if it can be removed from the problem and how to go about doing this. There are many nice post-processing tools available for analyzing the output data of SIESTA, such as the VisIt visualization package and the Poincare plotting suite contained inside.

While this paper focused more closely on the computational aspects of the SIESTA code and some post-processing techniques, future papers (including the PhD thesis) will have much more of the physics that is under study and is being resolved by SIESTA. In particular, it would be interesting to study the Alfvén eigenspectrum that can be obtained from the eigenvalues and eigenvectors of the Hessian of the simulation. By
comparing the spectrum for an unconverged equilibrium that is output from VMEC with a converged equilibrium from SIESTA, the effects of the presence of islands on the Alfvén spectrum can be analyzed. This analysis will be done in detail in the future for tokamaks and stellarators.
Appendix A

HSX hill and well equilibria

Due to the strong helical shaping of HSX, many poloidal and toroidal modes are required to run a SIESTA simulation on the machine. In order to obtain convergence with such a large system, more effort needs to be put into the parallelized implementation of the SIESTA code. While an accurate analysis cannot yet be done on HSX, some early results have been obtained that show island chains developing in both the hill and well configurations.

In the 6.2% hill case, a large $\nu = 4/4$ island chain has been predicted and shown to occur in the vacuum configuration. Looking at the iota profile for this case shows the $4/4$ resonance at a normalized radius of $\rho \approx 0.65$ for the fixed boundary VMEC run that the results are displayed for.
Running a SIESTA simulation with 20 toroidal modes resulted in the following equilibrium pressure contours in which the 4/4 island chain has opened up. The horizontal axis is the poloidal angle and the vertical axis is the normalized radius. All of the following pressure plots have these axes.
Figure A.2: Pressure contours for 6.2% hill in HSX using 20 toroidal modes

Running a 6.2% hill simulation with 10 toroidal modes resulted in the following pressure contours:
In the 6.2% well case, a smaller $\iota = 8/7$ island chain has been predicted and shown to occur in the vacuum configuration. Looking at the iota profile for this case shows the $8/7$ resonance at a normalized radius of $\rho \approx 0.78$ for the fixed boundary VMEC run that the results are displayed for.
Running a SIESTA simulation with 20 toroidal modes resulted in the following equilibrium pressure contours in which the 8/7 island chain has opened up.
As can be seen from the above plots, SIESTA was able to recover the qualitative structure of the island chains in finite-beta simulations. Once the code is better adapted to the study of machines such as HSX, better calculations can be performed with more accurate results.
Bibliography


