manual

code development issues and future physics applications.

contents

1.1 Outline

This document is intended to organise the different potentially valuable improvements to the SPEC code, which could make it more robust, faster, and increase its capabilities. The document is divided in two categories. In Sec.1.2, Numerical Improvements: independent improvements that are of numerical importance but have no added physics value *per se*, although they may allow new or better physics investigations. In Sec.1.3, research topics that could be addressed with the code, either in its present form or after the completion of one or more topics listed in Sec.1.2.

1.2 Numerical Improvements

1.2.1 Downloading SPEC to `Git`

This should be straight forward and easy. SRH will work on this. Estimated date of completion 06/2017.

1.2.2 Compile code with `GCC` for error checking

1.2.3 Profile code with `gprof` to find inefficient lines of code

1.2.4 Run code with `Valgrind` to identify memory leaks

1.2.5 De-NAG-ification

This will take time . . . It is not, at present, crucial, as we all have access to NAG.

1.2.6 Revision of spectral-constraints

This is bit of a mess. All the mathematics is standard, and all that is required is for someone to calmly go through lots of algebra. This task should be high priority, as SRH suspects that the spectral constraints as presently enforced result in an ill-conditioned force vector, which means that the code is overly sensitive to the initial guess and does not converge robustly. Potential speed improvements are tremendous.

1.2.7 Extension to arbitrary toroidal angle

This can further reduce the required Fourier resolution, and so this can reduce the computation. SRH is particularly interested in this as it will allow for exotic configurations (knots, figure-8, etc.) that cannot presently be computed.

1.2.8 Exploit symmetry of the metric

This is easy, but somewhat tedious. Take a look at ma00aa to see what is required. Potential speed improvement is considerable.

1.2.9 Exploit symmetry of "local" Beltrami matrices

This is easy. Take a look at matrix, which constructs the Beltrami matrices, and mp00ac, which performs the inversion. Potential speed improvement is considerable.

1.2.10 Exploit block tri-diagonal structure of "global" linearized force balance matrix

This requires an efficient subroutine. SRH believes that Hirshman constructed such a routine [S.P. Hirshman *et al.* (2010)[1]]. The potential speed improvement is tremendous. See newton for where the tri-diagonal, linearized force-balance matrix is inverted.

1.2.11 Enforce Helicity constraint

This will allow investigation of different, arguably more-physical classes of equilibria. See ma02aa

1.2.12 Establish test-cases

A suite of test cases should be constructed, with different geometries etc., that run fast, and that can be benchmarked to machine precision.

1.2.13 Verify free-boundary

This is almost complete. The virtual casing routines need to be investigated and made more efficient. The virtual casing routine in slab geometry needs revision (because of an integral over an infinite domain).

1.2.14 Enforcement current "profile"

i.e., adjust μ's, fluxes and/or rotational transform to obtain desired current profile (without singular currents). This should be a reasonably simple task. An additional routine is required to iterate on the helicity multipliers etc. as required *after* the local Beltrami fields have been calculated and *before* the global force balance iterations proceed.

1.2.15 Interpret eigenvectors and eigenvalues of Hessian

This is already completed: see hesian. For toroidal geometry there is a complication; namely that the hessian matrix includes the derivatives of the spectral constraints. For Cartesian geometry, it is ready to go. SRH will begin writing a paper on the stability of slab MRxMHD equilibria.

1.3 Physics Applications

1.3.1 Calculate high-resolution equilibria

e.g. W7-X. requires: Sec.1.2.8, Sec.1.2.9, and other improvements that can make the code faster at high Fourier resolution

1.3.2 Calculate equilibria by conserving helicity and fluxes

Applications to saturated island studies, sawteeth, etc. requires: Sec.1.2.11

1.3.3 Calculate free-boundary stellarator equilibria

to predict scrape-off-layer (SOL) topologies and β-limits. requires: Sec.1.2.13

1.3.4 Evaluate stability of MRxMHD equilibria

perhaps starting from simplest system (slab tearing). requires: Sec.1.2.15

1.4 Revision of coordinate singularity: axisymmetric; polar coordinates;

1. Consider a general, magnetic vector potential given in Cartesian coordinates,

$$\mathbf{A} = A_x \nabla x + A_y \nabla y + A_z \nabla z + \nabla g \tag{1}$$

where A_x, A_y, A_z, and the as-yet-arbitrary gauge function, g, are regular at $(x,y)=(0,0)$, i.e. they can be expanded as a Taylor series, e.g.

$$A_x = \sum_{i,j} \alpha_{i,j} x^i y^j, \qquad A_y = \sum_{i,j} \beta_{i,j} x^i y^j, \qquad A_z = \sum_{i,j} \gamma_{i,j} x^i y^j, \qquad g = \sum_{i,j} \delta_{i,j} x^i y^j, \tag{2}$$

for small x and small y.

2. Note that we have restricted attention to the "axisymmetric" case, as there is no dependence on z.

3. The 'polar' coordinate transformation,

$$\begin{aligned} x &= r\cos\theta, \\ y &= r\sin\theta, \\ z &= \zeta, \end{aligned} \tag{3}$$

[1]S.P. Hirshman, K.S. Purumalla *et al.*, J. Comput. Phys. **229**, 6392 (2010)

induces the vector transformation

$$\begin{array}{rclcll} \nabla x & = & \cos\theta\,\nabla r & - & r\sin\theta\,\nabla\theta & , \\ \nabla y & = & \sin\theta\,\nabla r & + & r\cos\theta\,\nabla\theta & , \\ \nabla z & = & & & \nabla\zeta & . \end{array} \tag{4}$$

4. By repeated applications of the double-angle formula, the expressions for A_x, A_y and g can be cast as functions of (r, θ),

$$A_x = \sum_m r^m [a_{m,0} + a_{m,1}\, r^2 + a_{m,2}\, r^4 + \ldots] \sin(m\theta), \tag{5}$$

$$A_y = \sum_m r^m [b_{m,0} + b_{m,1}\, r^2 + b_{m,2}\, r^4 + \ldots] \cos(m\theta), \tag{6}$$

$$A_z = \sum_m r^m [c_{m,0} + c_{m,1}\, r^2 + c_{m,2}\, r^4 + \ldots] \cos(m\theta), \tag{7}$$

$$g = \sum_m r^m [g_{m,0} + g_{m,1}\, r^2 + g_{m,2}\, r^4 + \ldots] \sin(m\theta), \tag{8}$$

where attention is restricted to stellarator symmetric geometry, but similar expressions hold for the non-stellarator symmetric terms.

5. Collecting these expressions, the vector potential can be expressed

$$\mathbf{A} = A_r \nabla r + A_\theta \nabla\theta + A_\zeta \nabla\zeta + \partial_r g\, \nabla r + \partial_\theta g\, \nabla\theta, \tag{9}$$

where

$$\begin{array}{rclcccccccccl} A_r & = & r^0 & [& (& & & b_{0,0} & + & g_{1,0} &) + (\ldots)r^2 + (\ldots)r^4 + \ldots &] & \sin\theta \\ & + & r^1 & [& (& a_{1,0}/2 & + & b_{1,0}/2 & + & 2g_{2,0} &) + (\ldots)r^2 + (\ldots)r^4 + \ldots &] & \sin 2\theta \\ & + & r^2 & [& (& a_{2,0}/2 & + & b_{2,0}/2 & + & 3g_{3,0} &) + (\ldots)r^2 + (\ldots)r^4 + \ldots &] & \sin 3\theta \\ & + & \ldots \end{array} \tag{10}$$

(Note: Mathematica was used to perform the algebraic manipulations, and the relevant notebook is included as part of the SPEC CVS repository.)

6. There is precisely enough gauge freedom so that we may choose $A_r = 0$. For example, the choice

$$\begin{array}{rclcccccccl} g_{1,0} & = & - & & & & b_{0,0} & & & & , \\ g_{2,0} & = & - & (& a_{1,0}/2 & + & b_{1,0}/2 &) & / & 2 & , \\ g_{3,0} & = & - & (& a_{2,0}/2 & + & b_{2,0}/2 &) & / & 3 & , \\ \ldots & = & & \ldots \end{array} \tag{11}$$

eliminates the lowest order r dependence in each harmonic.

7. By working through the algebra (again, using Mathematica) the expressions for A_θ and A_ζ become

$$A_\theta = r^2 f_0(\rho) + r^3 f_1(\rho)\cos(\theta) + r^4 f_2(\rho)\cos(2\theta) + r^5 f_3(\rho)\cos(3\theta) + \ldots \tag{12}$$

$$A_\zeta = g_0(\rho) + r^1 g_1(\rho)\cos(\theta) + r^2 g_2(\rho)\cos(2\theta) + r^3 g_3(\rho)\cos(3\theta) + \ldots \tag{13}$$

where $\rho \equiv r^2$ and the $f_m(\rho)$ and $g_m(\rho)$ are abitrary polynomials in ρ. [The expression for A_ζ is unchanged from Eqn.(7).]

1.4.1 somewhat generally, . . .

1. For stellarator-symmetric configurations,

$$\mathbf{A} = \sum_{m,n} A_{\theta,m,n} \cos(m\theta - n\zeta)\nabla\theta + \sum_{m,n} A_{\zeta,m,n} \cos(m\theta - n\zeta)\nabla\zeta, \tag{14}$$

where now the dependence on ζ is included, and the angles are arbitrary.

2. The near-origin behaviour of A_θ and A_ζ given in Eqn.(12) and Eqn.(13) are flippantly generalized to

$$A_{\theta,m,n} = r^{m+2} f_{m,n}(\rho), \tag{15}$$

$$A_{\zeta,m,n} = r^m \quad g_{m,n}(\rho), \tag{16}$$

where the $f_{m,n}(\rho)$ and $g_{m,n}(\rho)$ are arbitrary polynomials in ρ.

3. Additional gauge freedom can be exploited: including an additional gauge term ∇h where h only depends on ζ, e.g.

$$h(\zeta) = h_{0,0}\,\zeta + \sum h_{0,n}\sin(-n\zeta), \tag{17}$$

does not change the magnetic field and does not change any of the above discussion.

4. The representation for the $A_{\theta,m,n}$ does not change, but we must clarify that Eqn.(16) holds for only the $m \neq 0$ harmonics:

$$A_{\zeta,m,n} \quad = \quad r^m \quad g_{m,n}(\rho), \quad \text{for } m \neq 0. \tag{18}$$

5. For the $m = 0$, $n \neq 0$ harmonics of A_ζ, including the additional gauge gives $A_{\zeta,0,n} = g_{0,n}(\rho) + n\,h_{0,n}$. Recall that $g_{0,n}(\rho) = g_{0,n,0} + g_{0,n,1}\rho + g_{0,n,2}\rho^2 + \ldots$, and we can choose $h_{0,n} = -g_{0,n,0}/n$ to obtain

$$A_{\zeta,m,n} \quad = \quad r^m \quad g_{m,n}(\rho), \quad \text{for } m = 0,\ n \neq 0, \text{ with } g_{m,n}(0) = 0. \tag{19}$$

6. For the $m = 0$, $n = 0$ harmonic of A_ζ, we have $A_{\zeta,0,0} = g_{0,0}(\rho) + h_{0,0}$. Similarly, choose $h_{0,0} = -g_{0,n,0}$ to obtain

$$A_{\zeta,m,m} \quad = \quad r^m \quad g_{m,n}(\rho), \quad \text{for } m = 0,\ n = 0, \text{ with } g_{m,n}(0) = 0. \tag{20}$$

7. To simplify the algorithmic implementation of these conditions, we shall introduce a ‘regularization’ factor, $\rho^{m/2} = r^m$.

8. Note that the representation for $A_{\theta,m,n}$ given in Eqn.(15), with an arbitrary polynomial $f_{m,n}(\rho) = f_{m,n,0} + f_{m,n,1}\rho + f_{m,n,2}\rho^2 + \ldots$, is equivalent to $A_{\theta,m,n} = \rho^{m/2}\alpha_{m,n}(\rho)$ where $\alpha_{m,n}(\rho)$ is an arbitrary polynomial with the constraint $\alpha_{m,n}(0) = 0$.

9. We can write the vector potential as

$$A_{\theta,m,n} \quad = \quad \rho^{m/2}\alpha_{m,n}(\rho), \quad \text{with } \alpha_{m,n}(0) = 0 \text{ for all } (m,n), \tag{21}$$
$$A_{\zeta,m,n} \quad = \quad \rho^{m/2}\beta_{m,n}(\rho), \quad \text{with } \beta_{m,n}(0) = 0 \text{ for } m = 0. \tag{22}$$

1.4.2 non-stellarator symmetric terms

1. Just guessing, for the non-stellarator-symmetric configurations,

$$A_{\theta,m,n} \quad = \quad \rho^{m/2}\alpha_{m,n}(\rho), \quad \text{with } \alpha_{m,n}(0) = 0 \text{ for all } (m,n), \tag{23}$$
$$A_{\zeta,m,n} \quad = \quad \rho^{m/2}\beta_{m,n}(\rho), \quad \text{with } \beta_{m,n}(0) = 0 \text{ for } m = 0. \tag{24}$$

manual.h last modified on 7-07-20 15:11:44.00; SPEC subroutines;