casing

Constructs the field created by the plasma currents, at an arbitrary, external location using virtual casing.

[called by: bnorml.]

contents

1.1 theory and numerics

1. Required inputs to this subroutine are the geometry of the plasma boundary,

$$\mathbf{x}(\theta,\zeta) \equiv x(\theta,\zeta)\mathbf{i} + y(\theta,\zeta)\mathbf{j} + z(\theta,\zeta)\mathbf{k}, \tag{1}$$

and the tangential field on this boundary,

$$\mathbf{B}_s = B^\theta \mathbf{e}_\theta + B^\zeta \mathbf{e}_\zeta, \tag{2}$$

where θ and ζ are arbitrary poloidal and toroidal angles, and $\mathbf{e}_\theta \equiv \partial\mathbf{x}/\partial\theta$, $\mathbf{e}_\zeta \equiv \partial\mathbf{x}/\partial\zeta$. This routine assumes that the plasma boundary is a flux surface, i.e. $\mathbf{B}\cdot\mathbf{e}_\theta \times \mathbf{e}_\zeta = 0$.

2. The virtual casing principle [Shafranov & Zakharov (1972)[1], Lazerson (2012)[2], Hanson (2015)[3]] shows that the field outside/inside the plasma arising from plasma currents inside/outside the boundary is equivalent to the field generated by a surface current,

$$\mathbf{j} = \mathbf{B}_s \times \mathbf{n}, \tag{3}$$

where $\mathbf{n}$ is normal to the surface.

3. The field at some arbitrary point, $\bar{\mathbf{x}}$, created by this surface current is given by

$$\mathbf{B}(\bar{\mathbf{x}}) = \int_{\mathcal{S}} \frac{(\mathbf{B}_s \times d\mathbf{s}) \times \hat{\mathbf{r}}}{r^2}, \tag{4}$$

where $d\mathbf{s} \equiv \mathbf{e}_\theta \times \mathbf{e}_\zeta \; d\theta d\zeta$.

4. For ease of notation introduce

$$\mathbf{J} \;\; \equiv \;\; \mathbf{B}_s \times d\mathbf{s} = \alpha\,\mathbf{e}_\theta - \beta\,\mathbf{e}_\zeta, \tag{5}$$

where $\alpha \equiv B_\zeta = B^\theta g_{\theta\zeta} + B^\zeta g_{\zeta\zeta}$ and $\beta \equiv B_\theta = B^\theta g_{\theta\theta} + B^\zeta g_{\theta\zeta}$,

5. We may write in Cartesian coordinates $\mathbf{J} = j_x\,\mathbf{i} + j_y\,\mathbf{j} + j_z\,\mathbf{k}$, where

$$j_x = \alpha\, x_\theta - \beta\, x_\zeta \tag{6}$$
$$j_y = \alpha\, y_\theta - \beta\, y_\zeta \tag{7}$$
$$j_z = \alpha\, z_\theta - \beta\, z_\zeta. \tag{8}$$

6. Requiring that the current,

$$\mathbf{j} \;\; \equiv \;\; \nabla\times\mathbf{B} = \sqrt{g}^{-1}(\partial_\theta B_\zeta - \partial_\zeta B_\theta)\,\mathbf{e}_s + \sqrt{g}^{-1}(\partial_\zeta B_s - \partial_s B_\zeta)\,\mathbf{e}_\theta + \sqrt{g}^{-1}(\partial_s B_\theta - \partial_\theta B_s)\,\mathbf{e}_\zeta, \tag{9}$$

has no normal component to the surface, i.e. $\mathbf{j}\cdot\nabla s = 0$, we obtain the condition $\partial_\theta B_\zeta = \partial_\zeta B_\theta$, or $\partial_\theta \alpha = \partial_\zeta \beta$. In axisymmetric configurations, where $\partial_\zeta \beta = 0$, we must have $\partial_\theta \alpha = 0$.

7. The displacement from an arbitrary point, (X,Y,Z), to a point, (x,y,z), that lies on the surface is given

$$\mathbf{r} \equiv r_x\,\mathbf{i} + r_y\,\mathbf{j} + r_z\,\mathbf{k} = (X-x)\,\mathbf{i} + (Y-y)\,\mathbf{j} + (Z-z)\,\mathbf{k}. \tag{10}$$

[1] V.D. Shafranov & L.E. Zakharov, Nucl. Fusion **12**, 599 (1972)
[2] S.A. Lazerson, Plasma Phys. Control. Fusion **54**, 122002 (2012)
[3] J.D. Hanson, Plasma Phys. Control. Fusion **57**, 115006 (2015)

8. The components of the magnetic field produced by the surface current are then

$$B^x = \oint\!\!\oint d\theta d\zeta \;\; (j_y r_z - j_z r_y)/r^3, \tag{11}$$

$$B^y = \oint\!\!\oint d\theta d\zeta \;\; (j_z r_x - j_x r_z)/r^3, \tag{12}$$

$$B^z = \oint\!\!\oint d\theta d\zeta \;\; (j_x r_y - j_y r_x)/r^3 \tag{13}$$

9. When all is said and done, this routine calculates

$$\int_0^{2\pi} \int_0^{2\pi} \texttt{vcintegrand} \; d\theta d\zeta \tag{14}$$

for a given (X, Y, Z), where `vcintegrand` is given in Eqn.(16).

10. The surface integral is performed using NAG: D01EAF, which uses an adaptive subdivision strategy and also computes absolute error estimates. The absolute and relative accuracy required are provided by the input `vcasingtol`. The minimum number of function evaluations is provided by the input `vcasingits`.

1.2 calculation of integrand

1. An adaptive integration is used to compute the integrals. Consequently, the magnetic field tangential to the plasma boundary is required at an arbitrary point. This is computed, as always, from $\mathbf{B} = \nabla \times \mathbf{A}$, and this provides $\mathbf{B} = B^\theta \mathbf{e}_\theta + B^\zeta \mathbf{e}_\zeta$. (Recall that $B^s = 0$ by construction on the plasma boundary.) (It would be MUCH faster to only require the tangential field on a regular grid!!!)

2. Then, the metric elements $g_{\theta\theta}$, $g_{\theta\zeta}$ and $g_{\zeta\zeta}$ are computed. These are used to "lower" the components of the magnetic field, $\mathbf{B} = B_\theta \nabla\theta + B_\zeta \nabla\zeta$. (Please check why B_s is not computed. Is it because $B_s \nabla s \times \mathbf{n} = 0$?)

3. The distance between the "evaluate" point, (X, Y, Z), and the given point on the surface, (x, y, z) is computed.

4. If the computational boundary becomes too close to the plasma boundary, the distance is small and this causes problems for the numerics. I have tried to regularize this problem by introducing $\epsilon \equiv$`vcasingeps`. Let the "distance" be

$$D \equiv \sqrt{(X-x)^2 + (Y-y)^2 + (Z-Z)^2 + \epsilon^2}. \tag{15}$$

5. On taking the limit that $\epsilon \to 0$, the virtual casing integrand is

$$\texttt{vcintegrand} \equiv (B_x n_x + B_y n_y + B_z n_z)(1 + 3\epsilon^2/D^2)/D^3, \tag{16}$$

where the normal vector is $\mathbf{n} \equiv n_x \mathbf{i} + n_y \mathbf{j} + n_z \mathbf{k}$. The normal vector, `Nxyz`, to the computational boundary (which does not change) is computed in preset. This needs to be revised.