mp00ac

Solves for magnetic vector potential given $\boldsymbol{\mu} \equiv (\Delta\psi_t, \Delta\psi_p, \mu)^T$.

[called by: ma02aa.] [calls: packab, curent and tr00ab.]

contents

1.1 unpacking fluxes, helicity multiplier

1. The vector of "parameters", $\boldsymbol{\mu}$, is unpacked. (Recall that $\boldsymbol{\mu}$ was "packed" in ma02aa.) In the following, $\boldsymbol{\psi} \equiv (\Delta\psi_t, \Delta\psi_p)^T$.

1.2 construction of linear system

1. The equation $\nabla \times \mathbf{B} = \mu \mathbf{B}$ is cast as a matrix equation,

$$\mathcal{M} \cdot \mathbf{a} = \mathcal{R}, \tag{1}$$

where $\mathbf{a}$ represents the degrees-of-freedom in the magnetic vector potential, $\mathbf{a} \equiv \{A_{\theta,e,i,l}, A_{\zeta,e,i,l}, \ldots\}$.

2. The matrix $\mathcal{M}$ is constructed from $\mathcal{A} \equiv$`dMA` and $\mathcal{D} \equiv$`dMD`, which were constructed in matrix, according to

$$\mathcal{M} \equiv \mathcal{A} - \mu \mathcal{D}. \tag{2}$$

Note that in the vacuum region, $\mu = 0$, so $\mathcal{M}$ reduces to $\mathcal{M} \equiv \mathcal{A}$.

3. The construction of the vector $\mathcal{R}$ is as follows:

   i. if `Lcoordinatesingularity=T`, then

$$\mathcal{R} \equiv -\left(\mathcal{B} - \mu \mathcal{E}\right) \cdot \boldsymbol{\psi} \tag{3}$$

   ii. if `Lcoordinatesingularity=F` and `Lplasmaregion=T`, then

$$\mathcal{R} \equiv -\mathcal{B} \cdot \boldsymbol{\psi} \tag{4}$$

   iii. if `Lcoordinatesingularity=F` and `Lvacuumregion=T`, then

$$\mathcal{R} \equiv -\mathcal{G} - \mathcal{B} \cdot \boldsymbol{\psi} \tag{5}$$

The quantities $\mathcal{B} \equiv$`dMB`, $\mathcal{E} \equiv$`dME` and $\mathcal{G} \equiv$`dMG` are constructed in matrix.

1.3 solving linear system

It is not assumed that the linear system is positive definite. The `LAPACK` routine `DSYSVX` is used to solve the linear system.

1.4 unpacking, . . .

1. The magnetic degrees-of-freedom are unpacked by packab.

2. The error flag, `ImagneticOK`, is set that indicates if the Beltrami fields were successfully constructed.

1.5 construction of "constraint" function

1. The construction of the function $\mathbf{f}(\boldsymbol{\mu})$ is required so that iterative methods can be used to construct the Beltrami field consistent with the required constraints (e.g. on the enclosed fluxes, helicity, rotational-transform,. . .). See ma02aa for additional details.

1.5.1 plasma region

(a) For `Lcoordinatesingularity = T`, the returned function is:

$$\mathbf{f}(\mu,\Delta\psi_p) \equiv \left\{ \begin{array}{ccccccccc} (& 0 & , & 0 &)^T, & \text{if } \texttt{Lconstraint} & = & -1 \\ (& 0 & , & 0 &)^T, & \text{if } \texttt{Lconstraint} & = & 0 \\ (& ɿ(+1) - \texttt{iota(lvol)} & , & 0 &)^T, & \text{if } \texttt{Lconstraint} & = & 1 \\ (& ? & , & ? &)^T, & \text{if } \texttt{Lconstraint} & = & 2 \end{array} \right. \tag{6}$$

(b) For `Lcoordinatesingularity = F`, the returned function is:

$$\mathbf{f}(\mu,\Delta\psi_p) \equiv \left\{ \begin{array}{ccccccccc} (& 0 & , & 0 &)^T, & \text{if } \texttt{Lconstraint} & = & -1 \\ (& 0 & , & 0 &)^T, & \text{if } \texttt{Lconstraint} & = & 0 \\ (& ɿ(-1) - \texttt{oita(lvol-1)} & , & ɿ(+1) - \texttt{iota(lvol)} &)^T, & \text{if } \texttt{Lconstraint} & = & 1 \\ (& ? & , & ? &)^T, & \text{if } \texttt{Lconstraint} & = & 2 \end{array} \right. \tag{7}$$

1.5.2 vacuum region

(a) For the vacuum region, the returned function is:

$$\mathbf{f}(\Delta\psi_t,\Delta\psi_p) \equiv \left\{ \begin{array}{ccccccccc} (& 0 & , & 0 &)^T, & \text{if } \texttt{Lconstraint} & = & -1 \\ (& I - \texttt{curtor} & , & G - \texttt{curpol} &)^T, & \text{if } \texttt{Lconstraint} & = & 0 \\ (& ɿ(-1) - \texttt{oita(lvol-1)} & , & G - \texttt{curpol} &)^T, & \text{if } \texttt{Lconstraint} & = & 1 \\ (& ? & , & ? &)^T, & \text{if } \texttt{Lconstraint} & = & 2 \end{array} \right. \tag{8}$$

2. The rotational-transform, ɿ, is computed by tr00ab; and the enclosed currents, I and G, are computed by curent.

1.6 early termination

1. If $|\mathbf{f}| < \texttt{mupftol}$, then early termination is enforced (i.e., `iflag` is set to negative integer). (See ma02aa for details of how mp00ac is called iteratively.)