dforce

Calculates $\mathbf{F}(\mathbf{x})$, where $\mathbf{x} \equiv \{\text{geometry}\} \equiv \{R_{i,v}, Z_{i,v}\}$ and $\mathbf{F} \equiv [[p + B^2/2]] + \{\text{spectral constraints}\}$, and $\nabla \mathbf{F}$.

[called by: hesian, newton, pc00aa, pc00ab and xspech.]

[calls: packxi, ma00aa, matrix, ma02aa, lforce, volume, packab, tr00ab, coords and brcast.]

contents

1.1 unpacking

1. The geometrical degrees of freedom are represented as a vector, $\mathbf{x} \equiv \{R_{i,v}, Z_{i,v}\}$, where $i = 1,$ `mn` labels the Fourier harmonic and $v = 1,$ `Mvol`-1 is the interface label. This vector is "unpacked" using packxi. (Note that packxi also sets the coordinate axis, i.e. the $R_{i,0}$ and $Z_{i,0}$.)

1.2 parallelization over volumes

1. In each volume, `vvol = 1, Mvol`,
   (a) the logical array `ImagneticOK(vvol)` is set to `.false.`
   (b) the energy and helicity matrices, `dMA(0:NN,0:NN)`, `dMB(0:NN,0:2)`, etc. are allocated;
   (c) the volume-integrated metric arrays, `DToocc`, etc. are allocated;
   (d) calls ma00aa to compute the volume-integrated metric arrays;
   (e) calls matrix to construct the energy and helicity matrices;
   (f) calls ma02aa to solve for the magnetic fields consistent with the appropriate constraints, perhaps by iterating on mp00ac;
   (g) calls volume to compute the volume of the v-th region;
   (h) calls lforce to compute $p + B^2/2$ (and the spectral constraints if required) on the inner and outer interfaces;
   (i) the derivatives of the force-balance will also be computed if `LComputeDerivatives = 1`;
2. After the parallelization loop over the volumes, brcast is called to broadcast the required information.

1.3 broadcasting

1. The required quantities are broadcast by brcast.

1.4 construction of force

1. The force vector, $\mathbf{F}(\mathbf{x})$, is a combination of the pressure-imbalance Fourier harmonics, $[[p + B^2/2]]_{i,v}$, where i labels Fourier harmonic and v is the interface label:

$$F_{i,v} \equiv \left[(p_{v+1} + B^2_{i,v+1}/2) - (p_v + B^2_{i,v}/2) \right] \times \texttt{BBweight}_i, \tag{1}$$

where `BBweight(i)` is defined in preset; and the spectral condensation constraints,

$$F_{i,v} \equiv I_{i,v} \times \texttt{epsilon} + S_{i,v,1} \times \texttt{sweight}_v - S_{i,v+1,0} \times \texttt{sweight}_{v+1}, \tag{2}$$

where the spectral condensation constraints, $I_{i,v}$, and the "star-like" poloidal angle constraints, $S_{i,v,\pm 1}$, are calculated and defined in lforce; and the $\texttt{sweight}_v$ are defined in preset.

1.5 construct derivatives of matrix equation

1. Matrix perturbation theory is used to compute the derivatives of the solution, i.e. the Beltrami fields, as the geometry of the interfaces changes:
   i. If `Lposdef = 0`, then NAG: F07ADF and NAG: F07AJF are used to invert the Beltrami matrix using an LU decomposition;
   ii. If `Lposdef = 1`, then NAG: F01ADF is used to invert the Beltrami matrix using a Cholesky decomposition;

1.6 extrapolation: planned redundant

1. The extrapolation constraint is $R_{j,1} = R_{j,2}\,\psi_1^{m/2}/\psi_2^{m/2}$. Combining this with the regularization factor for the geometry, i.e. $R_{j,i} = \psi_i^{m/2}\xi_{j,i}$, we obtain

$$\xi_{j,1} = R_{j,2}/\psi_2^{m/2}. \tag{3}$$

dforce.h last modified on 7-07-26 15:09:46.00; SPEC subroutines;