

**ma02aa**

Constructs Beltrami field in given volume consistent with flux, helicity, rotational-transform and/or parallel-current constraints.

[called by: [dforce](#).]

[calls: [packab](#), [df00ab](#), [mp00ac](#).]

**contents**

<b>1</b>	<b>ma02aa</b>	<b>1</b>
1.1	sequential quadratic programming . . . . .	1
1.2	Newton method . . . . .	1
1.3	“linear” method . . . . .	1
1.3.1	plasma volumes . . . . .	1
1.3.2	vacuum volume . . . . .	1
1.4	debugging: finite-difference confirmation of the derivatives of the rotational-transform . . . . .	2

**1.1 sequential quadratic programming**

1. Only relevant if LBsequad=T. See [LBeltrami](#) for details.
2. Documentation on the implementation of [NAG: E04UFF](#) is under construction.

**1.2 Newton method**

1. Only relevant if LBnewton=T. See [LBeltrami](#) for details.

**1.3 “linear” method**

1. Only relevant if LBlinear=T. See [LBeltrami](#) for details.
2. The quantity  $\mu$  is not treated as a “magnetic” degree-of-freedom equivalent to in the degrees-of-freedom in the magnetic vector potential (as it strictly should be, because it is a Lagrange multiplier introduced to enforce the helicity constraint).
3. In this case, the Beltrami equation,  $\nabla \times \mathbf{B} = \mu \mathbf{B}$ , is linear in the magnetic degrees-of-freedom.
4. The algorithm proceeds as follows:

**1.3.1 plasma volumes**

- (a) In addition to the enclosed toroidal flux,  $\Delta\psi_t$ , which is held constant in the plasma volumes, the Beltrami field in a given volume is assumed to be parameterized by  $\mu$  and  $\Delta\psi_p$ . (Note that  $\Delta\psi_p$  is not defined in a torus.)
- (b) These are “packed” into an array, e.g.  $\boldsymbol{\mu} \equiv (\mu, \Delta\psi_p)^T$ , so that standard library routines , e.g. [NAG: C05PCF](#), can be used to (iteratively) find the appropriately-constrained Beltrami solution, i.e.  $\mathbf{f}(\boldsymbol{\mu}) = 0$ .
- (c) The function  $\mathbf{f}(\boldsymbol{\mu})$ , which is computed by [mp00ac](#), is defined by the input parameter [Lconstraint](#):
  - i. If [Lconstraint](#) = -1, 0, then  $\mu$  is not varied and Nx dof=0.
  - ii. If [Lconstraint](#) = 1, then  $\mu$  is varied to satisfy the transform constraints; and Nx dof=1 in the simple torus and Nx dof=2 in the annular regions. (Note that in the “simple-torus” region, the enclosed poloidal flux  $\Delta\psi_p$  is not well-defined, and only  $\mu = \mu_1$  is varied in order to satisfy the transform constraint on the “outer” interface of that volume.)
  - iii. If [Lconstraint](#) = 2, then  $\mu = \mu_1$  is varied in order to satisfy the helicity constraint, and  $\Delta\psi_p = \mu_2$  is not varied, and Nx dof=1. (**under re-construction**)

**1.3.2 vacuum volume**

- (a) In the vacuum,  $\mu = 0$ , and the enclosed fluxes,  $\Delta\psi_t$  and  $\Delta\psi_p$ , are considered to parameterize the family of solutions. (These quantities may not be well-defined if  $\mathbf{B} \cdot \mathbf{n} \neq 0$  on the computational boundary.)
- (b) These are “packed” into an array,  $\boldsymbol{\mu} \equiv (\Delta\psi_t, \Delta\psi_p)^T$ , so that, as above, standard routines can be used to iteratively find the appropriately constrained solution, i.e.  $\mathbf{f}(\boldsymbol{\mu}) = 0$ .
- (c) The function  $\mathbf{f}(\boldsymbol{\mu})$ , which is computed by [mp00ac](#), is defined by the input parameter [Lconstraint](#):
  - i. If [Lconstraint](#) = -1, then  $\mu$  is not varied and Nx dof=0.
  - ii. If [Lconstraint](#) = 0, 2, then  $\mu$  is varied to satisfy the enclosed current constraints, and Nx dof=2.
  - iii. If [Lconstraint](#) = 1, then  $\mu$  is varied to satisfy the constraint on the transform on the inner boundary  $\equiv$  plasma boundary and the “linking” current, and Nx dof=2.

5. The Beltrami fields, and the rotational-transform and helicity etc. as required to determine the function  $\mathbf{f}(\boldsymbol{\mu})$  are calculated in [mp00ac](#).
6. This routine, [mp00ac](#), is called iteratively if `Nxdof > 1` via [NAG: C05PCF](#) to determine the appropriately constrained Beltrami field,  $\mathbf{B}_{\boldsymbol{\mu}}$ , so that  $\mathbf{f}(\boldsymbol{\mu}) = 0$ .
7. The input variables `mupftol` and `mupfits` control the required accuracy and maximum number of iterations.
8. If `Nxdof = 1`, then [mp00ac](#) is called only once to provide the Beltrami fields with the given value of  $\boldsymbol{\mu}$ .

#### 1.4 debugging: finite-difference confirmation of the derivatives of the rotational-transform

1. Note that the rotational-transform (if required) is calculated by [tr00ab](#), which is called by [mp00ac](#).
2. If `Lconstraint=1`, then [mp00ac](#) will ask [tr00ab](#) to compute the derivatives of the transform with respect to variations in the helicity-multiplier,  $\mu$ , and the enclosed poloidal-flux,  $\Delta\psi_p$ , so that [NAG: C05PCF](#) may more efficiently find the solution.
3. The required derivatives are

$$\frac{\partial_t}{\partial\mu} \tag{1}$$

$$\frac{\partial_t}{\partial\Delta\psi_p} \tag{2}$$

to improve the efficiency of the iterative search. A finite difference estimate of these derivatives is available; need `DEBUG`, `Lcheck=2` and `Lconstraint=1`.