global

Defines input namelists and global variables

[called by: .] [calls: .]

contents

1 global **1**
1.1 overview . . . 1
1.2 input list: `optimin` . . . 1

1.1 overview

1. Here, and elsewhere, input variables are shown in `red`. The input list is read from file and broadcast in al00aa.

1.2 input list: `optimin`

- `Isymmetric = 1` : enforce stellarator symmetry;
- `Itopology = 0` : selects knottedness of plasma:
  - if `Itopology=0`, the plasma is an unknot;
  - if `Itopology=1`, the plasma is a torus-knot, as described by `ptorusknot`, `qtorusknot`, `Rtorusaxis` and `atorusaxis`; note that a torus knot can also be described as an arbitrary knot, so this option will soon be deleted in favour of `Itopology=2`;
  - if `Itopology=2`, the plasma is an arbitrary knot; the parameters describing the knot are read from file, rdknot;
- `ptorusknot = 2` : only used if `Itopology=1`;
- `qtorusknot = 3` : only used if `Itopology=1`;
- `Rtorusaxis = 2.000D-00` : only used if `Itopology=1`;
- `atorusaxis = 1.000D-00` : only used if `Itopology=1`;
- `knotwind = 0.500D-00` : radius of knotted winding surface; used in windsf;
- `knotsurf = 0.200D-00` : radius of knotted plasma boundary; used in plassf;
- `kspring = -1.000D-00` : only relevant if `Itopology` $= 2$; if `kspring` > 0, the geometry of the knot is "relaxed" to minimize the spring energy; used in rdknot;
- `trelax = 1.000D-03` : "distance" knot is relaxed, rdknot;
- `Nfp = 1` : field periodicity;
- `AddCurrent = 0` : can include additional currents that are not degrees-of-freedom; under construction; in the optimization process;
- `AddFactor = 1.000D-00` : arbitrary factor that is used in debugging;
- `Osymmetric = 1` : include self-symmetric coil passing through $(\theta, \zeta) = (0, 0)$;
- `Hsymmetric = 1` : include self-symmetric coil passing through $(\theta, \zeta) = (0, \pi)$;
- `Npairs = 32` : number of symmetric pairs of coils to be added per period;
- `NFourier = 1` : number of Fourier modes to describe each coil;
- `Mwinding = 1` : poloidal helical winding; see iccoil;
- `Nwinding = 0` : toroidal helical winding; see iccoil;
- `bstol = 1.000D-04` : tolerance in Biot-Savart integral; passed through to oculus:bs00aa;
- `bsnlimit = 10000` : max. number of iterations used in Biot-Savart integral; passed through to oculus:bs00aa;
- `Loptimize = 0` : to turn on optimization algorithm; the function to be optimized is described in ofunct; used in xoptim;

- `Wbnormal = -1.000D-00` : used in ofunct;
- `Wclength = -1.000D-00` : used in ofunct;
- `Wepotent = -1.000D-00` : used in ofunct;
- `Rcharge = 1.000D-00` : used in ofunct;
- `Scharge = 1.000D-00` : used in ofunct;
- `Wtorflux = 0.000D-00` : used in ofunct;
- `Tflux = 0.000D-00` : used in ofunct;
- `Nflux = 1` : used in ofunct;
- `Vzetaoff = 0` : to allow $\zeta_{i,0}$ to vary during optimization;
- `Vcurrents = 0` : to allow the coil currents to vary during optimization;
- `absacc = 1.000D-04` :
- `Lpoincare = 0` : to construct Poincaré plot;
  - if `Lpoincare` $= 1$, then the fieldline parameter is the cylindrical toroidal angle, and so B^{ϕ} must not equal zero;
  - if `Lpoincare` $= -1$, then the fieldline parameter is the length;
- `odetol = 1.000D-10` : o.d.e. integration tolerance;
- `Ppts = 100` : Poincaré points per trajectory;
- `Ptrj = 8` : Poincaré trajectories;
- `Rl = -1.000D-00` : starting points for Poincaré plot;
- `Zl = -1.000D-00` : starting points for Poincaré plot;
- `Ru = -1.000D-00` : starting points for Poincaré plot;
- `Zu = -1.000D-00` : starting points for Poincaré plot;
- `Ndcoils = 512` : discrete line segments for filamentary description of the coils (to be saved in `coils.ext`); used in hdfint;