Code FLOW user manual.

Contents

1 Introduction.

Back to top

The present document describes the code FLOW. References and source files are freely available on the code web page, www.me.rochester.edu/~guazzott/FLOW_main.html. Throughout this document, namelists names are written in namelist color, variable names in variable color, routine names in routine color and file names in file color.

In the solution of the modified Grad-Shafranov equation, FLOW sets the magnetic poloidal flux to 0 at the plasma edge (unless otherwise specified) and to be maximum on the magnetic axis. The maximum value of ψ is indicated with ψ_c throughout this document; ψ_c is calculated during the solution of the equilibrium problem.

2 Initial Constants.

Back to top

Several constants are specified in the namelist input_constants. As for geometrical constants:

- rmajor is the major radius of the torus.
- x_size and z_size are the dimensions of the computational grid (these are deactivated if grid_type is equal to -10, see below).
- rcenter is the geometrical horizontal center of the grid, different from rmajor if the grid is offset with respect to the geometrical center of the plasma. zcenter is the geometrical vertical center of the grid.

Other programming constants:

- eq_type specifies the closure to be used. 1 is for MHD, 3 for kinetic with toroidal flow only.
- eq3_opt is an option for the kinetic closure. If it is equal to 1, temperatures are assigned as free functions of psi, if equal to 2, pressures are assigned. The value is irrelevant if eq_type is not 3.
- numerical is a logical constant, specifying whether any function of ψ is assigned through a numerical input (i.e., tables). If numerical is set to .false., no function is assigned from a numerical input. Otherwise, some or all functions will be assigned from tables.
- broot is a constant specifying the type of equilibrium to be sought:

0 → "Transonic" equilibrium, with supersonic poloidal velocity at the edge and subsonic poloidal velocity in the center. Sub- and supersonic velocities are defined with respect to the poloidal "sound" speed, i.e. the poloidal magnetoslow speed.

1 → Subsonic poloidal flow. This option can also be used for equilibria without poloidal rotation.

2 → Supersonic poloidal flow.

3 → SuperAlfvénic poloidal flow (note: this option has not been used in a while, so it is not up to date).

4 → Subnsonic poloidal flow for Reversed Field Pinch equilibria.

5 → Transonic poloidal flow for Reversed Field Pinch equilibria.

- `mass` and `eV` are the ion mass and the elementary charge (set either to physical values or to 1).
- `me_ov_mi` and `pe_ov_p` are the ratio between electron and ion mass and between electron pressure and total pressure respectively. These are only used for bootstrap current calculation.

3 Algorithm Settings.

Back to top

The solution algorithm uses a red-black, multi-grid approach. The variables to be set are contained in the namelist input_solver. The following variables can be assigned from the input file:

- `n_min` is the number of grid points in each direction for the initial grid. (`n_min` -1) must be a multiple of 2.
- `n` is the number of grid points in each direction for the final grid. (`n` -1) must be a multiple of 2. In general, `n` $= 129$ is a good selection for standard equilibria. Obviously, it must be `n_min` $\leq$ `n`.
- `min_it` is the minimum number of iterations for each grid (usually set to 50).
- `max_it` is the maximum number of iterations for each grid (usually set from a few hundreds to ~ 2000).
- `accelerate` is a logical variable, controlling whether Chebishev acceleration has to be used in the SOR process.

- `fix_orp` is used as a constant relaxation parameter only if the variable accelerate is set to .false.

A criterion for convergence is specified in the file mgrid.f90 by the parameter eps, and in general should not be changed.

4 Boundary Conditions.

Back to top

Boundary Conditions are also controlled in the namelist input_solver.

Various different options are present, even though, essentially, two main algorithms ("flat" ψ and linear interpolation) are used in standard tokamak and tokamak-like equilibria. The choice is made assigning the value of the variable `bc_type`:

- `bc_type` $= 1 \rightarrow \psi$ is assigned to be 0 in all points outside the domain of integration (this is the least accurate option).
- `bc_type` $= 3 \rightarrow \psi$ is interpolated in the grid points immediately outside the computational domain.
- `bc_type` $= 4 \rightarrow$ `bc_type`$= 1$ is used for `nn`$<$`bc_switch`, `bc_type`$= 3$ is used for `nn`$\geq$`bc_switch` (`nn` is the grid resolution of the current grid in the multigrid process). This is the recommended option for tokamak-like equilibria.
- `bc_type` $= 5$ is used for LDX equilibria (ψ is assigned on the outer boundary and $\nabla\psi$ on the inner boundary).
- `bc_type` $= 6$ is used for LDX equilibria (ψ is assigned on both the outer and the inner boundary).
- `bc_type` $= 7$ is used for free-boundary calculations: $\psi = 0$ determines the plasma edge, ψ is assigned on a fixed boundary. The same algorithm selection as in `bc_type` $= 4$ are used.
- `bc_type` $= 8 \rightarrow$ like 7, but there is no need for a magnetic axis in the plasma.

Notice that the option `bc_type` $= 2$ has been removed.

As mentioned before, the variable `bc_switch` sets the resolution level where the code switches from simple, "flat" ψ boundary conditions to an interpolation algorithm (for the values of `bc_type` that allow it). This is done to avoid interpolation errors that could prevent convergence for coarse grids in case of sharp gradients near the edge. The recommended setting is `bc_switch` $= 65$.

5 Magnetic Field Data.

Back to top

The magnetic field data are contained in the namelist input_magnetic. The free function $F(\psi) = B_0(\psi)R_0$, controlling the toroidal field, is assigned according to the input described in this section.

B_φ in the vacuum is given by the variable b_phi_zero ($\rightarrow B_{\varphi 0}$ in equations in the rest of this document).

The variable F_opt controls the type of equation used for $F(\psi)$.

- If F_opt is equal to 0, then $B_\varphi = 0$.
- If F_opt is equal to 1, then

$$F(\psi) = F_{vacuum} + (F_{center} - F_{vacuum}) \left(\frac{\psi}{\psi_c}\right)^k .$$

  The ratio F_{center}/F_{vacuum} is assigned in the variable Fc_o_Fv. The exponent k (kappa) is assigned in the namelist as well.
- If F_opt is equal to 2, then

$$F(\psi) = \sqrt{F_{vacuum} - 2\eta_P \mu_0 R_0^2 P(\psi)};$$

  η_P is assigned in the variable eta_P.
- If F_opt is equal to 5, then the RFP toroidal field shape is used:

$$F(\psi) = F_{vacuum} \left[1 + \mu_{RFP} \left(\frac{\psi}{\psi_c} - 1 + \frac{(1 - \psi/\psi_c)^{k+1}}{k+1}\right)\right].$$

  $\mu_{RFP} \rightarrow$ mu_RFP.

Where needed, $F_{vacuum} \equiv B_{\varphi 0} R_0$. The variable mu_mag is the permeability of free space.

6 Flow Data.

Back to top

The flow data are contained in the namelist input_flow. Values for the sonic Mach numbers can be assigned arbitrarily. If eq_type = 3, the poloidal flow is automatically set equal to 0.

The default definition for the toroidal flow is given by:

$$M_\varphi(\psi) = M_\varphi^{MAX} \left(\frac{\psi}{\psi_c} + M_\varphi^{min}\right)^{\alpha_{M\varphi}} .$$

The constants $M_\varphi^{MAX} \equiv$ mach_phi_max, $\alpha_{M\varphi} \equiv$ alpha_mphi and $M_\varphi^{min} \equiv$ mphi_min are specified in the namelist.

The default definition for poloidal flow is given by:

$$M_\theta(\psi) = \begin{cases} M_\theta^e + (M_\theta^{MAX} - M_\theta^e)\left[\frac{2}{t}\frac{\psi}{\psi_c} - \left(\frac{\psi}{t\psi_c}\right)^2\right] & \text{if } \psi < t \\ \frac{M_\theta^{MAX}}{t^3}\left(2\psi - \frac{\psi}{\psi_c}\right)^2\left(2\frac{\psi}{\psi_c} - t\right) & \text{if } t < \psi < 2t \\ 0 & \text{if } \psi > 2t \end{cases},$$

where $M_\theta^{MAX} \rightarrow$ mach_theta_max, $M_\theta^e \rightarrow$ mach_theta_edge and $t \rightarrow$ t_mth. A similar definition, but with $t \rightarrow$ w_mth is used in the case of RFP equilibrium.

If a different definition for either $M_\varphi(\psi)$ or $M_\theta(\psi)$ is required, it must be specified in the functions mach_phi and mach_theta respectively. *Very important*: the first derivatives must be specified consistently in the function dmach_phidpsi and dmach_thetadpsi. An alternative is to specify the functions through numerical tables, as described in section 8.

7 Pressure and Density Profiles.

Back to top

Pressure and density are specified in the namelist input_p_d_profile. If the values in the namelist input_constants are in metric units, the values in namelist input_p_d_profile will be in metric units as well. The variable gamma defines the ratio of specific heats. In the isotropic case, pressure and density are assigned by the equations:

$$P(\psi) = P_{edge} + (P_{center} - P_{edge})\psi^\alpha \tag{1}$$

and

$$D(\psi) = D_{edge} + (D_{center} - D_{edge})\psi^{\alpha\rho}.$$

D_{center} is assigned in the corresponding variable dcenter, while for D_{edge} it is assigned $D_{edge} = D_{center}$ de_o_dc. Variable de_o_dc is assigned in the input. P_{center} is assigned through the variable beta_center via $P_{center} = 2\mu_0\beta_{center}/B_{\varphi 0}^2$. P_{edge} is derived from $P_{edge} = P_{center}$ pe_o_pc. Variable pe_o_pc is assigned in the input. The default expression for $P(\psi)$ holds for p_opt $\neq$4–10 (the other possibilities for $P(\psi)$ are used for specific cases described in function pofpsi) in file trans_solve.f90. Exponents $\alpha \rightarrow$ alpha and $\alpha_\rho \rightarrow$ alpha_rho must also be assigned.

If the system is anisotropic (eq_type = 3), either Quasi-Pressures or Quasi-Temperatures can be assigned. In the first case (eq3_opt = 2), both parallel and perpendicular pressures are assigned similarly to what is done in the isotropic case. Parallel and perpendicular input betas are therefore required. The edge values are obtained from qpee_o_qpec $= P_{\perp edge}/P_{\perp center}$ and qpae_o_qpac $= P_{\parallel edge}/P_{\parallel center}$.

If eq3_opt is equal to 1, parallel temperature is assigned as in (1), while perpendicular quasi-temperature is obtained from:

$$T_\perp(\psi) = T_\parallel(\psi)\frac{B_0(\psi)}{B_0(\psi) - \Theta(\psi)T_\parallel(\psi)},$$

where $B_0(\psi) \equiv F(\psi)/R_0$ and $\Theta(\psi) =$ theteps$B_0(\psi)/T_\parallel(\psi)$. $\Theta(\psi)$ is a measure of the anisotropy of the system. Parallel temperature is controlled by variables tparcenter and tpae_o_tpac.

8 Numerical Input.

Back to top

The namelist input_numerical specifies which free functions of ψ are assigned from numerical data, as summarized in the following table:

Flag	Function	File
numerical_n	$D(\psi)$	n.dat
numerical_p_iso	$P(\psi)$	p_iso.dat
numerical_p_par	$P_\parallel(\psi)$	p_par.dat
numerical_p_perp	$P_\perp(\psi)$	p_perp.dat
numerical_F	$R_0F(\psi)$	b0.dat
numerical_omega	$M_\varphi(\psi)$ OR $\Omega(\psi)$	omega.dat
numerical_mtheta	$M_\theta(\psi)$	mtheta.dat
numerical_psiprim	$\nabla\psi$ for bc_type=5	psiprim.dat

The variable omega_option is used to select whether the file omega.dat contains $M_\varphi(\psi)$ (for omega_option=1) or $\Omega(\psi)$ (for omega_option=2).

Each of the files listed above must contain a list of (ψ,function) values, with ψ ranging from 0 to 1, where 0 corresponds to the boundary and 1 to the magnetic axis. The points do not need to be equally spaced. Each file can have a different number of data points.

Again, if the variable numerical in the namelist input_constants is set to .false., no numerical input will be used, regardless of the content of the namelist input_numerical.

9 Gravity.

Back to top

The effect of a gravitational potential are included in FLOW, and controlled through the namelist input_gravity. Three variables are used to control the effect of gravity:

- gravity_type is set to 0 to neglect the effect of gravity, to 1 to include the effect of a point mass in the origin of the system of coordinates.
- G_gravity sets the value of the gravitational constant G. This is done to allow for different/normalized units to be used in the calculation.
- M_gravity sets the point mass originating the gravity field.

10 Plasma Shape and Triangularity.

Back to top

Plasma shape is controlled by the namelist input_triangularity. Several options are implemented in FLOW, and are still present mainly for back compatibility. In practice, three options are recommended:

- tri_type $= 0 \rightarrow$ no triangularity, the plasma is circular or elliptical. The exact shape is controlled by a_elps and b_elps, throught the equation
$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$$
(with $a \rightarrow$ a_elps, $b \rightarrow$ b_elps).
- tri_type $= 8 \rightarrow$ numerical input for plasma shape. The shape is entered as a table with $\{r(\theta), \theta\}$, where $r(\theta)$ is measured from the geometric center of the plasma (rmajor, 0). The angle θ is also measured from the geometric center of the plasma, with the outer midplane corresponding to $\theta = 0$. Data must contain at least a full loop, $0 \leq \theta \leq 2\pi$.
- tri_type $= 18 \rightarrow$ "standard" input for tokamak shapes: the contour (R, Z) is given by:
$$\begin{cases} R = R_0 + a\cos\left[\theta + \delta\sin(\theta)\right] \\ Z = \kappa a \sin(\theta), \end{cases}$$

where $R_0 \rightarrow$ `rmajor`, $a \rightarrow$ `a_elps`, $\kappa \rightarrow$ `k_ellipt`, $\delta \rightarrow$ `delta_up` / `delta_down` (the first one controls the triangularity of the plasma in the upper midplane, the second on in the lower midplane).

- Also, `tri_type` = -1 is useful for calculating free-boundary equilibria. This option corresponds to a rectangular boundary. Horizontal and vertical dimensions are controlled by variables `a_elps` and `b_elps`.

11 Output.

Back to top

While the code is running, the solution process is shown on the screen in Fig. 1. Once the run is completed, the final value of ψ_c is printed on the

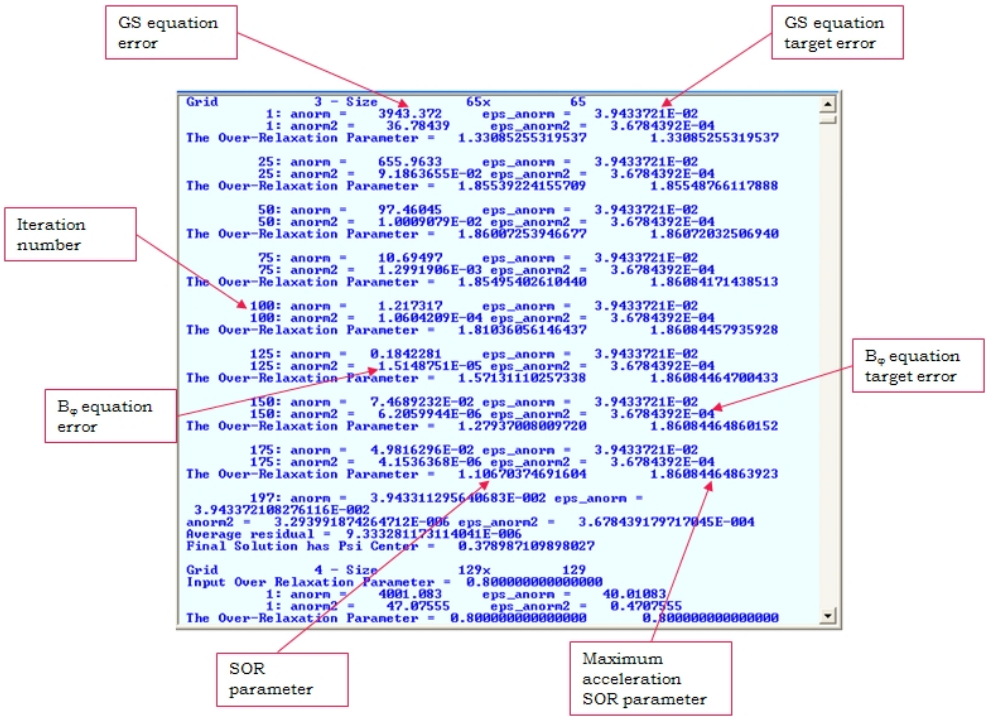

Figure 1: Running Output.

screen, together with the position of the magnetic axis. The output of the code consist in files containing the calculated values for each of the following quantities:

psi
rho
p
p_par
p_perp
beta
beta_par
beta_perp
Mach_Alfven poloidal
Mach_cusp
Mach_slow
Mach_phi (sonic Mach number)
v_phi
v_poloidal
v_r
v_z
B_phi
B_r
B_z
residual (solution error)
Temperature
T_par
T_perp
cs
csp
j_par
j_phi
j_x
j_z

For each quantity, two different files are saved:

1. a tecplot (.plt) file, containing the 2D output of the variable. The data are arranged as (R,Z,variable), with Z varying faster.
2. a .txt file, containing a line cut of the variable along the midplane.

Additional outputs are: bootstrap_R.plt, bootstrap.plt (containing bootstrap current calculation), magnetic_R.plt, magnetic_psi.plt (containing safety factor output), neoclass_res_R.plt, neoclass_res.plt (containing neoclassical resistivity calculation), trapped.plt

If variable write_all in namelist solver is set to .false., only psi and rho will be saved.

12 Other Input Variables.

Back to top

13 Miscellaneous Issues.

Back to top

13.1 Convergence Problems.

In some occasions, FLOW may converge slowly or not at all. Two different issues that may prevent convergence are known.

First, lack of convergence may be due to the input free functions. This may happen because the free functions are ill defined (i.e., there is no equilibrium defined by the input). Another possibility is that the free functions have large gradients, which make the solution procedure numerically unstable. There is no obvious solution for the first instance (other than changing the input). In the second case, it is advisable to smooth the input (in particular if it is assigned with numerical tables). Another useful tool is to use an underrelaxation solution process. This can be done by setting variable accelerate in input_solver to .false., and variable fix_orp to a value smaller than unity.

Second, boundary conditions may not converge in isolated points (this is only observed with bc_type equal to 3 or 4). The reason for this behavior has not yet been determined. In general, slightly changing the grid size (either one or both of x_size and z_size) will fix the problem.