Implementation of poloidal density variation in collision operator

September 8, 2017

These notes are concerned with the implementation of the `poloidalVariationInCollisionOperator` option in SFINCS version 3.

Φ_1 implementation in SFINCS

In a previous set of notes (Φ_1 *implementation*), poloidal density variation to the lowest order was implemented by linearizing around a poloidally varying Maxwellian f_0, rather than a flux-function Maxwellian f_M.

Poloidal variations can occur to lowest order if the potential varies on a flux-surface, in which case the lowest order distribution function becomes

$$f_0(\psi,\theta,\zeta) = f_M(\psi)e^{-Ze\Phi_1(\theta,\zeta)/T}. \tag{1}$$

[We will in the future suppress the ψ dependence, as SFINCS only solves for a single flux-surface.]

In the Φ_1 *implementation* notes, the equations were modified to linearize around this f_0 rather than f_M. However, the linearization in the collision operator was not treated, which is the subject of these notes.

Φ_1 implementation in the linearized collision operator

The linearized collision operator can be written as

$$C_{ab}^{L:f_0} = C_{ab}\{f_{a0}, f_{b0}\} + C_{ab}\{f_{a1}, f_{b0}\} + C_{ab}\{f_{a0}, f_{b1}\}, \tag{2}$$

where we use the superscript $L : f_0$ to indicate that the operator has been linearized around f_0 rather than f_M.

As f_0 and f_M have the same velocity space-structure, the terms in the above linearization can easily be expressed in terms of the terms in the linearization around f_M:

$$C_{ab}^{L:f_0} = C_{ab}\{f_{aM}, f_{bM}\}e^{-(Z_a/T_a+Z_b/T_b)e\Phi_1(\theta,\zeta)} + C_{ab}\{f_{a1}, f_{b0}\}e^{-Z_be\Phi_1(\theta,\zeta)/T_b} + C_{ab}\{f_{aM}, f_{b1}\}e^{-Z_ae\Phi_1(\theta,\zeta)/T_a}. \tag{3}$$

Effectively, from the explicit expressions for $C^{L:f_M}$, one can obtain $C^{L:f_0}$ by substituting the density of each species by:

$$n_a \to n_a e^{-\frac{Z_a e \Phi_1(\theta,\zeta)}{T_a}}. \tag{4}$$

Changes to the Jacobian

As the old collision operator only depend on f_1 (neglecting the temperature equilibration term $C_{ab}\{f_{aM}, f_{bM}\}$ for the moment), its contribution to the Jacobian was

$$J^{L:f_M} = \frac{\delta C^{L:f_M}}{\delta f_1}. \tag{5}$$

Furthermore, as the linearized collision operator is linear in f_1, we have

$$\frac{\delta C^{L:f_M}}{\delta f_1} = C^{L:f_M}[.], \tag{6}$$

where the notation $C^{L:f_M}[.]$ implies that the operator does not act on anything, i.e. the matrix used to represent $C^{L:f_M}[.]$ enters as is into the Jacobian.

In the system linearized around f_0, we have essentially the same f_1 dependence, if we substitute the density as in (**??**)

$$\frac{\delta C^{L:f_0}}{\delta f_1} = C^{L:f_M}[.] \left\{ n_a \to n_a e^{-\frac{Z_a e \Phi_1(\theta,\zeta)}{T_a}} \right\}, \tag{7}$$

however, as Φ_1 is an unknown, we get new non-zero elements in the Jacobian due to $\frac{\delta C^{L:f_0}}{\delta \Phi_1}$. As $C^{L:f_0}$ only depends on Φ_1 through $n_a e^{-\frac{Z_a e \Phi_1(\theta,\zeta)}{T_a}}$, we can also get the new expressions through as a density substitution

$$n_a e^{-\frac{Z_a e \Phi_1}{T_a}} \to -\frac{Z_a e}{T_a} n_a e^{-\frac{Z_a e \Phi_1}{T_a}}. \tag{8}$$

Thus

$$\frac{\delta C^{L:f_0}}{\delta \Phi_1} = C^{L:f_M}[f_1] \left\{ n_a \to -\frac{Z_a e}{T_a} n_a e^{-\frac{Z_a e \Phi_1}{T_a}} \right\}. \tag{9}$$

Note that the collision operator here acts on f_1.

In addition, if we include temperature equilibration, we get an additional contribtution to the Jacobian

$$\frac{\delta C^{L:f_0}}{\delta \Phi_1} = -e \frac{Z_a T_b + Z_b T_a}{T_a T_b} e^{\left(-\frac{Z_a T_b + Z_b T_a}{T_a T_b} e \Phi_1\right)} C_{ab}\{f_{aM}, f_{bM}\}. \tag{10}$$

This term, like all the other in this document, is merely an extra factor to an expression already calculated in the code.

Changes to the code

The collision operator is linearized around f_0 if `poloidalVariationInCollisionOperator` is set to true in the input file. This switch adds the extra factor $e^{-\frac{Z_a e\Phi_1}{T_a}}$ to the densities in the collision operator as calculated in the code, and reuses the calculations to evaluate $\frac{\delta C^{L:f_0}}{\delta \Phi_1}$. If `includeTemperatureEquilibrationTerm` is set to true, the contribution from this term is also included in the Jacobian.