|  |  |
| --- | --- |
| December 30, 2014 | Matt Landreman |

# Technical Documentation for version 3 of SFINCS

## Introduction

In this document, we detail the equations implemented in SFINCS: the Stellarator Fokker-Planck Iterative Neoclassical Conservative Solver. The normalizations and input and output quantities are also defined. We use SI units unless noted otherwise.

Differences between version 3 and previous versions of SFINCS include the following:

|  |  |
| --- | --- |
| **multiSpecies version** | **Version 3** |
| Adiabatic  assumed. | Full nonlinear effect of  can be included |
| Radial  drift is neglected. | Radial  drift can be included. |
| Magnetic drifts only appear in the inhomogeneous drive term. | Magnetic drifts acting on the perturbed  can also be included. |
| Boozer coordinates assumed. | Any flux coordinates can be used. |

## Kinetic equation

We begin with the drift-kinetic equation (19) of Hazeltine, Plasma Physics 15, 77 (1973):



where  is the full gyro-averaged distribution function,  denotes species,  is the collision operator, ,



is the total energy,  is the charge in units of the proton charge ,  is the mass,  is the gyrofrequency (which is negative for electrons with ), , , and . Subscripts on partial derivatives indicate quantities held fixed in differentiation. The drifts in are



and



In ,  is the electrostatic potential, which in general will vary on a flux surface.

For the slow neoclassical problem we wish to solve, we immediately drop all time derivative terms in except for , which represents an effect of the inductive electric field (if one is present). There is often no inductive electric field in a stellarator, but we retain this term for comparison with other codes such as DKES. In such codes, only the contribution from the leading-order Maxwellian



is retained in this inductive term, and we make the same approximation. In ,  denotes the leading order temperature, and  is the leading order density. Also, following appendix C of Landreman & Ernst, PPCF 54, 115006 (2012), we take the electromagnetic gauge to be chosen so that



where  denotes a flux surface average. Thus, our kinetic equation is reduced to

.

We next make the following changes:

* A source/sink term  is introduced, which is sometimes necessary to permit solvability, as discussed in the 2014 Phys. Plasmas paper on SFINCS.
* The  term in is neglected. Dropping this term is justified since  and only the non-Maxwellian part of the distribution function  contributes to , so overall this term has magnitude , which should be quite small.
* We replace  in the parallel magnetic drift. Since the parallel drift is smaller than the parallel streaming term  by a factor , this change should be safe. This change allows the magnetic drifts to be written as a curl,



As shown in Appendix B of Landreman & Ernst, PPCF 54, 115006 (2012), this form of the magnetic drifts is convenient for obtaining conservation laws.

With these changes, our kinetic equation becomes



where



is the drift-kinetic operator, and

.

## System of equations

For numerical solution, we will use coordinates  where , , and . We also choose a specific form for the source:

.

In this case, can be written



In the numerical approach proposed here, we solve together with the following equations:

 (quasineutrality)







where

.

Notice the full  is used everywhere in the system of equations except in the radial derivative in . In this term we must substitute  for , since otherwise the code would need to be 5D instead of 4D, and 5D is not feasible. For the same reason, everywhere in  that  appears, it will be approximated by .

In ,  is the collision operator linearized about a stationary Maxwellian; it could be either , or a Maxwellian with the actual poloidally-varying density  could be used. The more challenging latter approach might be important for high-Z impurities; we should think about how large we expect the impurity density asymmetry on a flux surface to be.

The profiles  and  represent additional particle and heat transport of species  required for a steady-state solution to exist.

The scalar quantity  that appears in is a sort of Lagrange multiplier which may be needed numerically to allow to be satisfied. Flux-surface-averaging and noting , then the code should always find  if the inputs  satisfy quasineutrality.

The unknowns in the system - are



Therefore, the number of degrees of freedom is



where  is the number of degrees of freedom in the coordinate  upon discretization. As you can verify, the number of scalar equations in - is the same as the number of unknowns , so the system is “square”. If we had not included , , or , the system would not be square. For comparison, the number of degrees of freedom in SFINCS presently is , so does not represent much of an increase.

Although - are linear in the unknowns, is nonlinear in the unknowns. One type of nonlinearity occurs in collisionless terms , such as the parallel acceleration term. If the collision operator is linearized about the actual density rather than , then the collision term would be nonlinear as well. The system of equations is solved by Newton’s method, evaluating the Jacobian analytically. The first iteration would begin with the initial values , , , , . Since the nonlinear terms are quadratic in the unknowns, analytic differentiation of is feasible.

## Magnetic geometry

In the nonlinear version of SFINCS, we allow the magnetic geometry to be specified in any coordinates  satisfying  where  is the toroidal flux, as long as physical quantities are periodic in  and . Boozer coordinates can be used, as can VMEC coordinates. We will not assume the field lines are straight in the SFINCS coordinates, and in principle one could specify a magnetic field with  (although any MHD equilibrium will have ). We will write the kinetic equation in terms of the following components of :



and the inverse Jacobian

.

In terms of the components , we have



and

.

Notice that for any quantity , we have



and

.

To evaluate the components of the magnetic drifts, we will need to know the components of . These components may be evaluated using and a couple of vector identities. The results are

,

,

.

In , we have introduced the parameter  which is either 0 or 1. The user sets  by setting forceZeroRadialCurrent = true in the input file. The rationale for introducing the parameter  is that some choices of coordinates (though not Boozer coordinates) permit there to be a nonzero radial current in the magnetic equilibrium, in which case a Maxwellian ends up giving a net radial current:



which seems like it could be pathological. We prove this fact following ???. Thus, the parameter  allows the user to determine whether the nonzero radial current in the magnetic geometry has any significant effect on the SFINCS output.

We will need to evaluate the components of the drifts. The components of the  drift are

,

,

,

where we have made the required approximation . For the DKES version of the  drift, we choose to only replace  in the  terms, not the terms involving . We make this choice because DKES does not include . Thus,

 (unchanged),

,

.

To evaluate the magnetic drifts, we first re-write as

.

Using with -, the components of the magnetic drifts are

,

,

.

We will need to evaluate flux surface averages. For any quantity , the flux surface average is



where

.

In the case that Boozer coordinates are used, then

,

where  is the rotational transform with  the safety factor, and

.

where , ,  is the poloidal current outside the flux surface, and  is the toroidal current inside the flux surface. The product of with gives

.

The components of  become



Note that  is assumed whenever Boozer coordinates are used, and indeed  and  mean vanishes. Thus,  is irrelevant for Boozer coordinates.

If we are given non-Boozer coordinates, it is possible to compute  and  without much effort. Consider a closed curve at fixed  and  with . Applying Ampere’s Law,



where  is the poloidal current outside the flux surface. In the general coordinate system, we can write the length element along this curve as . Then using , the left-hand side of can be written

.

But this same analysis can be applied in Boozer coordinates, in which case the left-hand side is . (The integration curves are somewhat different in the 2 coordinate systems, since different  coordinates are held fixed, but the results of the integration must be the same according to Ampere’s Law since the curves enclose the same current.) Thus,



where the left-hand side is evaluated in the non-Boozer system. The left-hand side must evidently be independent of , since the right-hand side is -independent. We choose to average over  since doing so may reduce numerical error slightly. Thus,

.

We can repeat the analysis for a poloidally closed curve instead of a toroidally closed curve, giving

.

## Trajectory coefficients

Now let us evaluate the coefficients  in in terms of the coordinates used internally in SFINCS. First, we consider the spatial coefficients:





and



where we have introduced  or 1. In the bottom lines of -, the black terms are the terms included in the linear version of SFINCS. The blue terms in - are associated with the poloidal or toroidal magnetic drifts, and are turned on and off by the parameter includeSmallLinearMagneticDriftTerms, corresponding to  or 0. The magenta terms in are *linear* terms that are unique to the nonlinear version of SFINCS, because they involve  but will multiply only the known quantity . The green terms in - are nonlinear, since they involve  and will multiply derivatives of the full . (Note that  and  each contain one linear and one nonlinear term.)

Next, let us evaluate . To do so, we first write

,

so

.

Plugging in -, several cancellations occur, leaving



where we have introduced , , and , which are either 0 or 1. In , the cyan and red terms are present in the linear version of SFINCS, although these terms are slightly simpler in the linear version because  is assumed in that version. In the linear version of SFINCS, the cyan and red terms both act on the Maxwellian contribution . In the linear version of SFINCS, the cyan term does not act on the perturbation , while in the nonlinear version, we do allow this cyan term to act on the full  if includeTemperatureGradientTerms = true, corresponding to . The red terms in always act on the Maxwellian term . The red terms may or may not also act on the perturbation ; this action is turned on and off by the parameter includeXDotTerm, corresponding to  or 0, as in the linear version of SFINCS. The blue terms in are small, as they involve both the poloidal/toroidal magnetic drifts and . These blue terms are turned on and off by the parameter includeSmallNonlinearMagneticDriftTerms, corresponding to  or 0. Both the blue and green terms are nonlinear, since they involve  and will multiply .

Note that the last 2 green terms in are , so these terms may be important for heavy impurities.

Next, let us evaluate . To do so, we first write



and so

.

Into we substitute - and . Several cancellations occur so that the result has no factors of  in a denominator (which is important for the Legendre expansion we are about to use). After some algebra, the results simplify to



The  factors have been inserted here to reflect the actual implementation in the code for turning various terms on and off, and are not “derived” from previous equations. The first line of , in black, is the familiar mirror force term, also present in the linear version of SFINCS. The next line, in red, is also present in linear SFINCS, and may be turned on or off with the includeElectricFieldTermInXDot parameter, corresponding to  or 0. The next two lines, in green, are nonlinear, as they involve  and will multiply . The last line of , in blue, is linear but is not present in the linear version of SFINCS, as it arises from the magnetic drifts.

## Normalizations

Let’s suppose we are given , , , , , , , , , , and  where  is the normalized toroidal flux. The flux at the last closed flux surface is , so the dimensional flux  is related to  by . The input quantities are specified in terms of some species-independent dimensions  (e.g. eV),  (e.g. 1020/m3),  (e.g. kV),  (e.g. T),  (e.g. m), and  (typically the proton or deuteron mass). In other words, the quantities we are actually given are

,

,

,

,

,

,

,

,

,

,

,

,

,

,



,

and

.

Notice , and so



for any flux function . If Boozer coordinates are used, then we also have

,

.

It will be useful to define the following combinations of normalization constants:

,



(which resembles ; note that  has a factor of  in Gaussian units),

,

and a normalized collisionality



where  is the dimensional collisionality at the reference parameters:

 (SI units),

 (Gaussian units)

We assume  has the same value for all species. Notice that



with no 2 inside the square root.

As in the multi-species version of SFINCS, we define a normalized distribution function  as follows:

.

Notice this normalization is the same for each species.

The kinetic equation for each species is made dimensionless by multiplying through by

.

This normalization is slightly different from the linear version of SFINCS. The associated normalized trajectory coefficients may then be written as



so the kinetic equation is



where



is the normalized collision operator and



denote the normalized sources. Applying our normalizations to -, , and , the normalized trajectory coefficients are









and



## Legendre discretization

SFINCS uses a collocation discretization in the , , and  coordinates, but a modal discretization in the  coordinate. In other words, the distribution function is known at certain grid points in , , and , but it is expanded as modes in . We employ the following modal expansion in terms of Legendre polynomials :

.

We discretize the kinetic equation by applying

.

To evaluate the various integrals that result, we use the orthogonality relation

,

as well as the following identities:

,





,



,

and

.

As a result, may be written



where



,



(Since  only acts on the Maxwellian, we only need to consider  in the identities -),







and



When the  term operates on the Maxwellian, we only care about the  terms:



,

,

,





,

,

and





## Collision operator

The total collision operator for species  is a sum of collision operators with each species:

.

The linearized Fokker-Planck collision operator for each pair of species may be written

,

where the Lorentz part of the collision term is



with

,

,

.

The energy scattering contribution is



where

.

The field term is



where the potentials are defined by



and

.

We write the field term as



where







The Poisson equations that define the potentials are (for Legendre mode )



.

Let us define





so the defining equations for the potentials become



.

Next, recall that in the kinetic equation , we need to evaluate



where  is defined in . It is convenient to note



Expanding  as before,



where



The energy scattering component is



The diagonal term is



In the cross-species case, this term is no longer identical to the  term in energy scattering (as it is in the same-species case).

The  term in the collision operator is



Although in principle we would also be free to write



(i.e. replacing  in two places), the resulting expression is less convenient because we compute  on the  grid, and so it is easier to differentiate with respect to .

The  collision term is



## Radial coordinates

There are many flux surface labels which may be used as radial coordinates. Examples include the toroidal flux (divided by ) normalized to the reference values:

,

the toroidal flux normalized to the flux at the last closed magnetic surface:

,

a normalized effective minor radius:



(where  is called  in the code), and an effective minor radius:



where  is any effective minor radius of the last closed flux surface and

.

Put another way, there are  main options: the flux or the square root of the flux, normalized either by the “Bar” quantities or normalized by values at the last closed flux surface. The input gradients (i.e. gradients of density, temperature, and electrostatic potential) may be specified as derivatives with respect to any of these 4 radial coordinates; you choose between these options using the parameter gradientInputScheme. Whichever choice you make, sfincs converts all gradients to  derivatives internally for the main computations. For any quantity , this conversion is done using

,

,

.

A similar conversion is done for output fluxes. For example, the code first computes the radial particle flux with respect to the coordinate , i.e. . Fluxes with respect to other radial coordinates are then computed using

,

,

.

## Output quantities

In the definitions below, recall that  and  are flux functions - the average density and temperature on the flux surface, not the *total* density and temperature.

### Flux surface averages:

For any quantity , the flux surface average can be computed from



where

.

One output quantity we give for convenience is

.

### Density

SFINCS returns the density carried in :

.

We also return the total density

.

Another quantity saved is



which should be nearly zero, within roundoff error or so.

### Pressure perturbation

SFINCS also returns the pressure in , normalized to the reference pressure :



The total pressure is also saved:



Upon flux surface averaging, we obtain



Perhaps I should also save pressures with the velocity shifted by the mean flow?

I could also save the temperature perturbations.

### Flow

Several quantities related to the parallel fluid flow are available in the output file:

,

,

,

and

.

I might want to eventually also include the Mach number computed from the local thermal speed.

Both numerical and analytic calculations often employ a weighted average flow, such as . In SFINCS, we save several variants of this average parallel velocity:

,

,

,

.

### Parallel current

Closely related to the parallel flow is the parallel current

.

The weighted average parallel current  is often considered, and it is available in the code as

.

In order to give an average parallel current without the overall scaling by , we also provide two other related outputs:



and



### General comments on radial fluxes

The radial fluxes have the form



where  is a flux surface label. In SFINCS we give fluxes for 4 possible flux surface labels: , , , and , indicated by the suffix , , , or  to the variable names in the code and in the .h5 output file. Below, we illustrate calculations for the specific case . (While we give the flux-surface-averaged fluxes with respect to multiple radial coordinates, for the “beforeSurfaceIntegral” versions of the fluxes we only give results for the psiHat radial coordinate.)

For the flux of each quantity (particles, momentum, and energy), and for each radial coordinate, one can compute the flux associated with the leading-order Maxwellian or with the full distribution function. One can also compute the flux associated with the radial magnetic drift, with the radial  drift, and with the total . In SFINCS the following fluxes are recorded, using the particle flux as an example:

|  |  |
| --- | --- |
| **Variable name** | **Definition, up to normalization** |
|  | i.e. just the Maxwellian. This quantity should be 0 to high precision unless there is a radial current in the magnetic equilibrium. |
|  | i.e., the full distribution function |
|  | i.e. just the Maxwellian. This flux is typically nonzero, unlike . |
|  | i.e., the full distribution function |
|  |  |
|  | i.e., the full distribution function. In a sense this is the most complete radial flux available in sfincs. |

When  is false, only the first 2 fluxes in the table are computed.

The rationale for defining  is that (at least for ) this quantity (computed only when ) gives the particle flux that would be expected if , such as the flux computed in the older versions of sfincs. This relationship is proved in separate notes ?????

For a similar reason, we also define one other version of the heat flux:

For calculating the fluxes below, it is useful to note that from we know



and

.

We will also repeatedly use

.

### Particle fluxes

The contribution to the particle flux from the magnetic drift which we save in SFINCS is defined to be

.

Using , , , and , we write as



where



Using and the identities



and



we obtain



Repeating the last few steps for the radial  drift instead of the magnetic drift,

,

we find



where



The quantities  and  in the nonlinear version of SFINCS are equivalent (i.e. normalized in the same way) to the quantities  and  in the linear multispecies version.

### Momentum flux

Particle and heat fluxes are usually more important than momentum fluxes, but just in case it ever turns out to be useful, we do compute and save a momentum flux in SFINCS. Since the parallel flow moments are often computed with an extra factor of  in the flux surface average, we do the same here for the momentum fluxes. (This factor of  was not included in previous linear versions of SFINCS!) The momentum flux due to magnetic drifts that we save is

,

i.e., compared to the particle flux there is an extra factor of

.

Thus, in place of - we have



where



(Differences from the particle flux are highlighted in red.) To evaluate the  integrals we can use



and

,

so becomes



Similarly, the contribution to the momentum flux from the radial  drift is

.

Furthermore,



where



### Heat flux

The contribution to the heat flux from the magnetic drift which we save in SFINCS is defined to be

.

In other words, there is an extra factor of



compared to the particle flux. Thus, instead of and we have



and



where differences from the particle flux are given in red. Similarly, the contribution to the radial heat flux from the radial  drift is

,

given by



where



The quantities  and  in the nonlinear version of SFINCS are equivalent (i.e. normalized in the same way) to the quantities  and  in the linear multispecies version.

## Transport matrix

.

 \















