GKW how and why

A.G. Peeters, R. Buchholz, Y. Camenen, F.J. Casson,
S. Grosshauser, W.A. Hornsby, P. Manas, P. Migliano,
M. Siccinio, A.P. Snodin, D. Strintzi, T. Sung,
G. Szepesi, D. Zarzoso

Updated in 2015, work in progress. [1]

[1]Copyright © 2007-2015, by named authors.

Preface

This documentation comes with the GKW code. It is largely based on our 2009 paper in Computer Physics Communications Ref. [1], but contains additional material and descriptions of additional features that the code has aquired since that paper. In this manual GKW is presented with the aim of documenting the exact equations solved, presenting the essential benchmarks which document the proper numerical solution, and explaining the code structure and installation. This document is evolving along with the code, and currency and completeness presently take precedence over integration and polish of all sections. GKW developers are strongly urged to update this document (doc.tex) to reflect changes made in the code repository.

What is GKW?

Gyrokinetic Workshop (GKW) is a code for the simulation of microinstabilities and turbulence in a magnetically confined plasma. The gyrokinetic model is is five dimensional partial differential equation and its solution requires a massively parallel approach. GKW is written in Fortran 95 and has been parallelised using MPI, which enables it to scale well up to 32768+ cores. The code was initially developed at the University of Warwick in 2007 through the extension of the linear code LINART [2], and can be used both in the linear as well as in the nonlinear regime. In 2016 it has been renamed from 'Gyrokinetics at Warwick' to 'Gyrokinetic Workshop', to reflect the nonlocality of its developers and users. The code has been freely available online since 2010, and has been hosted at https://bitbucket.org/gkw/gkw, since 2015, where it is actively maintained.

Can I use GKW?

You can obtain and use GKW for any purpose with minimal restrictions. Essentially you can make any changes to the source, and are not obliged to inform us or communicate these changes back to us (although we would welcome it if you did). Furthermore, you are free to redistribute the code to others under the terms of the license (see the LICENSE file), which is the GNU General Public License version 3.

We politely request that the effort made by us is recognised. In any publication that uses the results of GKW, either directly or indirectly, you should cite the CPC paper [1]. (If you want to be kind to us you could additionally cite the original LINART paper Ref. [2] and/or the first paper in which GKW has been used Ref. [3]).

References

1. A.G. Peeters, Y. Camenen, F.J. Casson, W.A. Hornsby, A.P. Snodin, D. Strintzi, G. Szepesi, Computer Physics Communications, **180**, 2650, (2009)

2. A.G. Peeters, D. Strintzi, Phys. Plasmas **11**, 3748, (2004)

3. A.G. Peeters, C. Angioni, D. Strintzi, Phys. Rev. Lett. **98**, 265003, (2007)

What does GKW do?

GKW aims at solving the turbulent transport problem arising in tokamak plasmas. It solves the gyrokinetic (the kinetic Vlasov equation averaged over the fast gyro-motion) equation on a fixed grid in the 5-dimensional space using a combination of finite difference and pseudo spectral methods. The solution of the kinetic equations governing this problem is an active area of research, with many groups worldwide contributing. The two approaches used are Particle in Cell (PIC) codes [4, 5, 6, 7, 8, 9, 10], and Vlasov codes [11, 12, 13, 14, 15]. GKW obviously falls in the latter category, and reaches a standard that can be expected from a modern Vlasov code, i.e. it includes kinetic electrons, a general description of the geometry with a coupling to a MHD equilibrium solver, electro-magnetic effects, collisions, and the effect of ExB shearing. In addition the physics of toroidal plasma rotation is included through a formulation of the equations in the co-moving system. The physics of rotation is currently an active area of research and many of the applications of GKW have concentrated on the transport of toroidal momentum [3, 17, 18, 19, 20, 21, 22, 23, 24], the influence of rotation on other transport channels [25, 26], and the interaction between turbulence and islands [27, 28].

Code summary

Licensing provisions: GNU GPLv3

Programming language: Fortran 95

RAM: $\sim$ 128MB-1GB for a linear run; 25GB for typical nonlinear kinetic run (30 million grid points).

Number of processors used: 1, although the program can efficiently utilise 4096+ processors (spectral), depending on problem and available computer. 128 processors is reasonable for a typical nonlinear kinetic run on the latest x86-64 machines. The nonspectral scheme can efficiently scale to 32768+ processors.

External routines/libraries: None required, although the functionality of the program is somewhat limited without a MPI implementation (preferably MPI-2) and the FFTW3 library. In addition, the non-spectral and implicit schemes require the UMFPACK and AMD libraries, and the eigenvalue solver requires the SLEPc and PETSc libraries.

Standard solution method: Pseudo-spectral and finite difference with explicit time integration.

Running time: (On recent x86-64 hardware) $\sim$5 minutes for a short linear problem; 4000 hours for typical nonlinear kinetic run.

Geometry Interface: The MHD equilibrium code CHEASE [42] is used for the general geometry calculations. This code has been developed in CRPP Lausanne and is not distributed together with GKW, but can be downloaded separately. The geometry module of GKW is based on the version 7.3 of CHEASE, which includes the output for Hamada coordinates.

Contents

Chapter 1

Gyrokinetic theory

1.1 The gyrokinetic framework

The starting point of the equation for the evolution of the distribution function is the gyrokinetic theory formulated in Refs [29, 30, 31, 32]. In GKW these equations are solved up to first order in the Larmor radius over the major radius $\rho_* \ll 1$ (with the exception of the polarisation which enters in the field equations). In order to determine which of the terms in the original formulation of the theory (which is accurate up to second order in ρ_*) must be kept, we will here briefly discuss the ordering. More details can be found in Ref. [20].

A thermal velocity ($v_{\rm th}$) and a major radius (R) are used to normalise the length and time scales. For the gradients we assume

$$R\nabla_\parallel \approx 1 \qquad R\nabla_\perp \approx 1/\rho_*, \tag{1.1}$$

i.e. the solution is assumed to have a length scale of the order of the system size along the field, but can have a length scale of the order of the Larmor radius perpendicular to the magnetic field. This difference in length scales means that the convecting velocities must be evaluated to a different order parallel and perpendicular to the field. The velocity along the field is to be evaluated in lowest order only (i.e. of the order of $v_{\rm th}$), but the perpendicular velocity must be evaluated up to first order (order $\rho_* v_{\rm th}$). In GKW the perturbations in the electro-static potential ϕ as well as the vector potential $\mathbf{A}$ are retained. For the latter it is, however, assumed that

$$\mathbf{A} = A_\parallel \mathbf{b}, \tag{1.2}$$

where $\mathbf{b}$ is the unit vector in the direction of the unperturbed magnetic field. The assumption above means that no compressional effects of the magnetic field are retained in the description. Compressional effects are currently implemented in the code but are not yet described here. Note that the vector potential above is the perturbed vector potential not to be confused with the one connected with the equilibrium magnetic field. The equations will be formulated in the co-moving system of a toroidally rotating plasma. The $E \times B$ drift velocity connected with this toroidal rotation is of the order of the thermal velocity in the laboratory frame, but the transformation to the co-moving frame removes this largest order $E \times B$ velocity from the equations [20, 51]. The transformation to the co-moving frame also introduces a background centrifugal potential Φ. The perturbed potential ϕ, perturbed vector potential $A_\parallel$, and background potential Φ in the co-moving frame can then be taken to be of the order

$$\phi \approx \frac{T}{e}\rho_* \qquad A_\parallel \approx \frac{T}{e v_{\rm th}}\rho_* \qquad \Phi \approx \frac{T}{e} \tag{1.3}$$

where T is the temperature, and e is the elementary charge. Since Φ is an equilibrium quantity, the length scale of any variation is assumed large compared to the Larmor radius, whilst the perturbed quantities may have perpendicular gradient length scales on the order of the Larmor radius. The field gradients can then be shown to be of the same order, with $\nabla_\perp \phi \approx \nabla_\perp \Phi$. From these assumptions it can be derived that the $E \times B$ velocity is of the order $\rho_* v_{\rm th}$.

Gyro-centre coordinates $(\mathbf{X}, \mu, v_\parallel)$ are used, with $\mathbf{X}$ being the gyro-centre position, $v_\parallel$ the parallel (to the magnetic field) velocity, and μ the magnetic moment $\mu = mv_\perp^2/2B$. Here $v_\perp$ is the velocity component perpendicular to the equilibrium magnetic field $\mathbf{B}$, m is the particle mass and B is the magnetic field strength. The magnetic moment is an invariant of motion, but the parallel velocity changes due to the electric field acceleration and mirror force. It turns out that for consistency one needs to keep both the lowest as well as the first order (in ρ_*) contribution to the acceleration.

Finally, the approximation known as the δf approximation is employed. This approximation assumes that the perturbed distribution f is much smaller than the background distribution F, but has a perpendicular length scale of the order of the Larmor radius whereas the background varies on the length scale of the system size:

$$f \approx \rho_* F \qquad \frac{\partial f}{\partial v_\parallel} \approx \rho_* \frac{\partial F}{\partial v_\parallel} \qquad \nabla f \approx \nabla F. \tag{1.4}$$

The ordering assumption above removes the 'velocity non-linearity' (combinations of the electro-magnetic fields and the velocity space derivative of the perturbed distribution) from the evolution equation of the perturbed distribution. The disadvantage is that strict energy conservation no longer applies.

With the help of these approximations the gyrokinetic equation of Refs. [29, 31, 32] can be rewritten.

1.2 Lagrangian

(This section is not in the CPC paper which instead refers to Ref. [20] which contains the full derivation with centrifugal effects.) (CHECK CONSISTENCY)

The starting point is the Lagrangian

$$\gamma = \left(\frac{e}{c}\mathbf{A} + \frac{e}{c}A_\parallel \mathbf{b} + m\mathbf{v}\right) \cdot d\mathbf{x} - \left(\frac{m}{2}v^2 + e\phi\right) dt \tag{1.5}$$

The coordinates are transformed into guiding center coordinates. The electrostatic guiding center Lagrangian is known, and can be found in Hahm 1988 [30]. For the magnetic perturbations we take that

$$A_\parallel(\mathbf{X} + \rho)\mathbf{b} \cdot d(\mathbf{X} + \rho) \tag{1.6}$$

where $\rho = \rho(\mathbf{X}, \mu, \theta)$. All the derivatives of ρ will give vectors perpendicular to the magnetic field. Thus if there is not a perpendicular magnetic potential perturbation, the only term that survives is

$$A_\parallel \mathbf{b} \cdot d\mathbf{X}$$

Then the guiding center Lagrangian is

$$\gamma = \left(\frac{e}{c}\mathbf{A} + \frac{e}{c}A_\parallel \mathbf{b} + mv_\parallel \mathbf{b}\right) \cdot d\mathbf{X} + \mu d\theta - \left(\frac{m}{2}v_\parallel^2 + \mu B + e\phi\right) dt \tag{1.7}$$

This Lagrangian is transformed in the gyrocenter coordinates with the Lie transforms. To the first order, if there are no perpendicular perturbations in the magnetic potential, we can just substitute the perturbations of the fields with the respective gyroaveraged quantities. This can also be derived rigorously with the Lie transforms. The new Lagrangian will be

$$\Gamma = \left(\frac{e}{c}\mathbf{A} + \frac{e}{c} < A_\parallel > \mathbf{b} + mv_\parallel \mathbf{b}\right) \cdot d\mathbf{X} + \mu d\theta - \left(\frac{m}{2}v_\parallel^2 + \mu B + e < \phi >\right) dt \tag{1.8}$$

where now $\mathbf{X}, \mu, v_\parallel, \theta$ are the gyrocenter coordinates. From this one can derive the equations of motion using the Euler Lagrange equations

$$\omega_{ij}\frac{dz^j}{dt} = \frac{\partial H}{\partial z^i} - \frac{\partial \gamma_i}{\partial t} \tag{1.9}$$

where H is the Hamiltonian and

$$\omega_{ij} = \frac{\partial \gamma_j}{\partial z^i} - \frac{\partial \gamma_i}{\partial z^j} \tag{1.10}$$

are the Lagrange brackets.

The Lagrange brackets that are of interest are

$$\omega_{X_i X_j} = \frac{e}{c} \epsilon_{ijk} \hat{B}_k \tag{1.11}$$

$$\omega_{\mathbf{X}U} = -m\mathbf{b} \tag{1.12}$$

where

$$\hat{\mathbf{B}} = \nabla \times \mathbf{A} + \nabla \times A_\parallel \mathbf{b} + \frac{mc}{e} v_\parallel \nabla \times \mathbf{b} \tag{1.13}$$

Choosing $i = U$ we get that

$$m\mathbf{b} \cdot \dot{\mathbf{X}} = m v_\parallel \tag{1.14}$$

Choosing $i = \mathbf{X}$ we get that

$$m\dot{v}_\parallel \mathbf{b} - \frac{e}{c} \dot{\mathbf{X}} \times \hat{\mathbf{B}} = -e\nabla < \phi > -\mu \nabla B - \frac{e}{c} \frac{\partial < A_\parallel >}{\partial t} \mathbf{b} \tag{1.15}$$

and taking that

$$\mathbf{E} = -\nabla \phi - \frac{1}{c} \frac{\partial < A_\parallel >}{\partial t} \mathbf{b} \tag{1.16}$$

we get that

$$m\dot{v}_\parallel \mathbf{b} - \frac{e}{c} \dot{\mathbf{X}} \times \hat{\mathbf{B}} = e\mathbf{E} - \mu \nabla B \tag{1.17}$$

We can cross this equation with $\mathbf{b}$ to derive the equation for $\dot{X}$. To derive the equation of GKW we have to approximate

$$\nabla \times (< A_\parallel > \mathbf{b}) \approx \nabla(< A_\parallel >) \times \mathbf{b} \tag{1.18}$$

Then the equation of $\dot{X}$ is.

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = v_\parallel \mathbf{b} + \mathbf{v}_D + \mathbf{v}_E + \mathbf{v}_{\delta B_\perp}, \tag{1.19}$$

To derive the equation for $v_\parallel$ we dot with $\dot{X}$. This gives

$$m v_\parallel \dot{v}_\parallel = \dot{\mathbf{X}} \cdot (e\mathbf{E} - \mu \nabla B) \tag{1.20}$$

1.3 The gyrokinetic equation

In a rigidly rotating frame, the evolution of equation for the gyro-centre position $\mathbf{X}$, and the parallel velocity are [20]

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = v_\parallel \mathbf{b} + \mathbf{v}_D + \mathbf{v}_\chi, \qquad m v_\parallel \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot \left[Ze\mathbf{E} - \mu \nabla B + m\Omega^2 R \nabla R \right], \qquad \frac{\mathrm{d}\mu}{\mathrm{d}t} = 0. \tag{1.21}$$

Here Z is the particle charge, R the local major radius, and Ω the frame rotation frequency. $\mathbf{E}$ is the gyro-averaged perturbed electric field plus the inertial electric field

$$\mathbf{E} = -\nabla\langle\phi\rangle - \frac{\partial\langle A_\parallel\rangle}{\partial t}\mathbf{b} - \nabla\langle\Phi\rangle, \tag{1.22}$$

where the angle brackets denote the gyro-average. Since Φ is an equilibrium quantity, the scale lengths of any variation are assumed large compared to the Larmor radius and we neglect the gyroaverage, taking $\Phi \approx \langle\Phi\rangle$. (The full details of the calculation of Φ are given in Sec. 4.1).

The velocities in Eq. 1.21 are from left to right: the parallel motion along the unperturbed field ($v_\parallel\mathbf{b}$), the drift motion due to the inhomogeneous field ($\mathbf{v}_D$), and

$$\mathbf{v}_\chi = \frac{\mathbf{b}\times\nabla\chi}{B} \qquad \text{with} \qquad \chi = \langle\phi\rangle - v_\parallel\langle A_\parallel\rangle, \tag{1.23}$$

which is the combination of the $E \times B$ velocity ($\mathbf{v}_E = \mathbf{b}\times\nabla\langle\phi\rangle/B$) and the parallel motion along the perturbed field line ($\mathbf{v}_{\delta B} = -\mathbf{b}\times\nabla v_\parallel\langle A_\parallel\rangle/B$). The drifts due to the inhomogeneous magnetic field and inertial terms can be written in the form

$$\mathbf{v}_D = \frac{1}{Ze}\left[\frac{mv_\parallel^2}{B} + \mu\right]\frac{\mathbf{B}\times\nabla B}{B^2} + \frac{mv_\parallel^2}{2ZeB}\beta'\mathbf{b}\times\nabla\psi + \frac{2mv_\parallel}{ZeB}\boldsymbol{\Omega}_\perp + \frac{1}{ZeB}\mathbf{b}\times\nabla\mathcal{E}_\Omega \tag{1.24}$$

In the equation above ψ is a radial coordinate (flux label) and $\boldsymbol{\Omega}_\perp$ is the angular (toroidal) rotation vector perpendicular to the field, i.e. $\boldsymbol{\Omega}_\perp = \boldsymbol{\Omega} - (\boldsymbol{\Omega}\cdot\mathbf{b})\mathbf{b}$. The first term on the right is the combination of the grad-B drift and curvature drift in the low beta approximation, whereas the second term, involving

$$\beta' = \frac{2\mu_0}{B^2}\frac{\partial p}{\partial\psi}, \tag{1.25}$$

is the correction to the curvature drift due to the modification of the equilibrium associated with the pressure gradient. The penultimate term is the Coriolis drift derived in Ref. [3] using the formulation of the gyrokinetic equations in the co-moving frame [35]. The formulation of Ref. [3] was extended in Refs. [20, 51] to include the centrifugal force, and the first centrifugal results were published in Refs. [26, 51]. The last term combines the centrifugal drift and background potential in the (species dependent) centrifugal energy

$$\mathcal{E}_\Omega = Ze\Phi - \frac{1}{2}m\Omega^2(R^2 - R_0^2) \tag{1.26}$$

where R_0 is the major radius of the point on the flux surface at which the densities are defined.

From the equations above the gyrokinetic equation in $(\mathbf{X}, v_\parallel, \mu)$ coordinates can be derived: [29, 20]

$$\frac{\partial f_{\rm tot}}{\partial t} + \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t}\cdot\nabla_\mu f_{\rm tot} + \frac{\mathrm{d}v_\parallel}{\mathrm{d}t}\frac{\partial f_{\rm tot}}{\partial v_\parallel} = 0. \tag{1.27}$$

The distribution function $f_{\rm tot}$ in this equation is split in a background F and a perturbed distribution f. As discussed in Section 1.1 the δf approximation will be employed. Consequently, the equation for f can be written in the form [20]

$$\frac{\partial f}{\partial t} + (v_\parallel\mathbf{b} + \mathbf{v}_D + \mathbf{v}_\chi)\cdot\nabla f - \frac{\mathbf{b}}{m}\cdot(\mu\nabla B + \nabla\mathcal{E}_\Omega)\frac{\partial f}{\partial v_\parallel} = S. \tag{1.28}$$

where S is determined by the background distribution function. The latter is assumed to be a Maxwellian (F_M), with temperature (T) and mean parallel velocity ($u_\parallel$) being functions of the radial coordinate only. In the case of a rotating plasma the density in the flux surface varies as $n(\theta) = n_{R_0}\exp(-\mathcal{E}_\Omega(\theta)/T)$ and is kept in the solution of the equilibrium equation [20].

$$F = F_M = \frac{n_{R_0}}{\pi^{3/2}v_{\rm th}^3}\exp\left[-\frac{(v_\parallel - (RB_t/B)\omega_\phi)^2 + 2\mu B/m}{v_{\rm th}^2} - \mathcal{E}_\Omega/T\right]. \tag{1.29}$$

Here $v_{\rm th} \equiv \sqrt{2T/m}$ is the thermal velocity (note that in this area of research the thermal velocity is usually defined without the $\sqrt{2}$ factor). The appearance of an explicit rotation velocity $u_\parallel$ in the Maxwellian may be confusing since the equations are formulated in the co-moving system in which the rotation vanishes. However, under experimental conditions the plasma rotation $\omega(\phi)$ has a radial gradient while the frame is chosen to rotate as a rigid body with a constant angular frequency Ω. Indeed a differential rotation of the frame would be impractical since the distance between two fixed points in the co-moving system would increase in time, i.e. the metric would be time dependent.

The local model constructed here chooses the rotation of the frame to be equal to the plasma rotation at one particular radius. Therefore ω_ϕ in the above Maxwellian is zero at the radial location considered, but has a finite radial gradient. This gradient will enter the equations as $\nabla\omega_\phi$. (for more details see Ref. [20]).

The term containing the derivative of the vector potential towards time presents a numerical difficulty. In the δf formalism it can, however, easily be combined with the time derivative of f. If one defines a 'new' distribution g

$$g = f + \frac{Ze}{T} v_\parallel \langle A_\parallel \rangle F_M, \tag{1.30}$$

and replaces the time derivative of f in the gyrokinetic equation with the time derivative of g, the cumbersome time derivative of $\langle A_\parallel\rangle$ is removed. Using the equations above one arrives at the following equation for the distribution function g

$$\frac{\partial g}{\partial t} + \mathbf{v}_\chi \cdot \nabla g + (v_\parallel \mathbf{b} + \mathbf{v}_D)\cdot\nabla f - \frac{\mathbf{b}}{m}\cdot(\mu\nabla B + \nabla\mathcal{E}_\Omega)\frac{\partial f}{\partial v_\parallel} = S. \tag{1.31}$$

where S is given by

$$S = -(\mathbf{v}_\chi + \mathbf{v}_D)\cdot\left[\frac{\nabla n_{R_0}}{n_{R_0}} + \left(\frac{v_\parallel^2}{v_{\rm th}^2} + \frac{(\mu B + \mathcal{E}_\Omega)}{T} - \frac{3}{2}\right)\frac{\nabla T}{T} + \frac{m v_\parallel}{T}\frac{RB_t}{B}\nabla\omega_\phi\right]F_M - \frac{Ze}{T}[v_\parallel \mathbf{b} + \mathbf{v}_D]\cdot\nabla\langle\phi\rangle F_M. \tag{1.32}$$

Note that both g and f appear in this equation. Since any form of dissipation on the field variables leads to a numerical instability [15], it is preferable to use f in any term that requires dissipation for stable numerical implementation. It should also be noted here that the Maxwellian is not an exact solution of the gyrokinetic equation in the limit of a zero perturbed electro-magnetic field. From the equation above it follows that S is nonzero due to the drifts connected with the magnetic field inhomogeneity in the gradients of density temperature and velocity. The solution of the evolution equation will determine the deviation away from the Maxwellian as a nonzero perturbed distribution f. This perturbation is of the order $\rho_* F_M$, and neglecting it as part of the background distribution is consistent with the ordering adopted. The drift term is $\mathbf{v}_D$ in the first term of the equation above is nevertheless implemented in GKW since the perturbation it generates is responsible for the neo-classical transport. All physics and diagnostic of the neo-classical fluxes have been implemented in the code, but the implementation has not been benchmarked and should therefore not be used.

To close the set of equations, one needs to calculate the potential and vector potential through the (low frequency ordering) Maxwell equations. These have been formulated in the literature, and will be given in Section 7.37 when the full set of equations in normalised form is discussed.

1.4 Fully electromagnetic gyrokinetic equation

In high beta tokamak plasmas the amplitude of the magnetic perturbations can become significant compared to the fluctuations of the electro-static potential. In such cases electro-magnetic effects should be taken into account in the Vlasov equation for an accurate gyrokinetic simulation. In this section the gyrokinetic Lagrangian, equations of motion, the Vlasov and Maxwell equations are summarized in the fully electro-magnetic, collisionless case in a rotating frame of reference. The detailed derivation of the equations can be found in derivation.tex.

The gyrokinetic Lagrangian is

$$\begin{aligned}\bar{\Gamma} &= \left(mv_\parallel \mathbf{b}(\mathbf{X}) + Ze\mathbf{A}_0(\mathbf{X}) + m\mathbf{u}_0 + Ze\langle\mathbf{A}_1\rangle(\mathbf{X})\right)\cdot \mathrm{d}\mathbf{X} + \frac{2\mu m}{Ze}\mathrm{d}\theta - \\ &\left(\frac{1}{2}m\left(v_\parallel^2 - u_0^2\right) + Ze\left(\Phi_0(\mathbf{X}) + \langle\phi\rangle(\mathbf{X})\right) + \mu\left(B_0(\mathbf{X}) + \langle B_{1\parallel}\rangle(\mathbf{X})\right)\right)\mathrm{d}t \end{aligned} \tag{1.33}$$

where $\langle\phi\rangle = J_0(\lambda)\phi$, $\langle\mathbf{A}_1\rangle = J_0(\lambda)\mathbf{A}_1$ and $\langle B_{1\parallel}\rangle = \hat{J}_1(\lambda)B_{1\parallel}$ are the gyroaveraged field perturbations. $\mathbf{b}$ is the unit vector along the equilibrium magnetic field, $\mathbf{u}_0$ is the velocity of the rotating frame of reference, and Φ_0 is the equilibrium electro-static potential due to the plasma rotation. The bar over $\bar{\Gamma}$ distinguishes the gyro-centre Lagrangian from its guiding-centre version.

The equations of motion can be derived from the Euler–Lagrange equations and are written as

$$\begin{aligned}\dot{\mathbf{X}} &= \mathbf{b}v_\parallel + \frac{mv_\parallel^2}{ZeB_0}\left(\nabla\times\mathbf{b}\right)_\perp + \frac{\langle\mathbf{B}_{1\perp}\rangle}{B_0}v_\parallel - \frac{1}{B_0}\langle\mathbf{E}\rangle\times\mathbf{b} + \\ &\frac{\mu}{ZeB_0}\nabla\left(B_0 + \langle B_{1\parallel}\rangle\right)\times\mathbf{b} + \frac{2mv_\parallel}{ZeB_0}\mathbf{\Omega}_\perp - \frac{mR\Omega^2}{ZeB_0}\nabla R\times\mathbf{b} \\ &= \mathbf{b}v_\parallel + \underbrace{\mathbf{v}_{\langle\mathbf{B}_{1\perp}\rangle} + \mathbf{v}_{\langle\mathbf{E}\rangle\times\mathbf{B}_0} + \mathbf{v}_{\nabla\langle B_{1\parallel}\rangle}}_{\mathbf{v}_\chi} + \underbrace{\mathbf{v}_{\mathrm{C}} + \mathbf{v}_{\nabla\mathbf{B}_0} + \mathbf{v}_{\mathrm{co}} + \mathbf{v}_{\mathrm{cf}}}_{\mathbf{v}_{\mathrm{D}}}\end{aligned} \tag{1.34}$$

where $\mathbf{\Omega}$ is the angular momentum vector associated with the rotation of the reference frame, and

$$\dot{v}_\parallel = \frac{\dot{\mathbf{X}}}{mv_\parallel}\cdot\left(Ze\langle\mathbf{E}\rangle - \mu\nabla(B_0 + \langle B_{1\parallel}\rangle) + \frac{1}{2}m\nabla u_0^2\right). \tag{1.35}$$

The quantity χ, related to the drifts due to perturbations of the fields, can be now written as

$$\chi = \underbrace{\Phi_0 + \langle\phi\rangle}_{\langle\Phi\rangle} - v_\parallel\langle A_{1\parallel}\rangle + \frac{\mu}{Ze}\langle B_{1\parallel}\rangle \tag{1.36}$$

and

$$\mathbf{v}_\chi = \frac{\mathbf{b}\times\nabla\chi}{B_0} = \mathbf{v}_{\langle\mathbf{B}_{1\perp}\rangle} + \mathbf{v}_{\langle\mathbf{E}\rangle\times\mathbf{B}_0} + \mathbf{v}_{\nabla\langle B_{1\parallel}\rangle}. \tag{1.37}$$

The Vlasov equation in presence of magnetic field compression takes the form

$$\frac{\partial g}{\partial t} + \mathbf{v}_\chi\cdot\nabla g + \left(v_\parallel\mathbf{b} + \mathbf{v}_{\mathrm{D}}\right)\cdot\nabla\delta f - \frac{\mu}{m}\mathbf{b}\cdot\left(\nabla\Phi_0 - mR\Omega^2\nabla R + \nabla B_0\right)\frac{\partial\delta f}{\partial v_\parallel} = S \tag{1.38}$$

with the source term being

$$S = -\left(\mathbf{v}_\chi + \mathbf{v}_{\mathrm{D}}\right)\cdot\nabla_p F_M + \frac{F_M}{T}\left(v_\parallel\mathbf{b} + \mathbf{v}_D\right)\cdot\left(-Ze\nabla\langle\phi\rangle - \mu\nabla\langle B_{1\parallel}\rangle\right) \tag{1.39}$$

where g is the modified distribution function

$$g = \delta f + \frac{Zev_\parallel}{T}\langle A_{1\parallel}\rangle F_{\mathrm{M}}. \tag{1.40}$$

Equations 1.38 and 1.39 show that there are four new terms appearing due to the inclusion of $B_{1\parallel}$. These are related to the convection of the perturbed distribution function in phase space due to the grad-B drift in the fluctuating magnetic field (non-linear), the same convection of the equilibrium distribution function (linear), and a couple of additional mirror force terms in the direction of the parallel and drift velocities (linear). The normalised form of these terms can be found in section 2.4. The former two is included in the modified definition of χ appearing in terms III and V (equations 2.263 and 2.265), the latter two are explicitly written in terms X and XI (equations 2.273 and 2.274).

1.4.1 Gyrokinetic field equations

The normalised gyrokinetic field equations in presence of $B_{1\parallel}$ are listed here. The normalizing assumptions can be found in section 2.1. The equations are written in Fourier-space, the Fourier components of the fields and the distribution function are denoted by $\hat{(.)}$. Γ_0 and Γ_1 are modified Bessel-functions, $b_{\rm sp}$ denotes their species dependent attributes in the Fourier-space.

The quasi-neutrality equation is

$$\sum_{\rm sp} Z_{\rm sp} n_{\rm R,sp} \left[2\pi B_{\rm N} \int J_0(k_\perp \rho_{\rm sp}) \hat{g}_{\rm N,sp} {\rm d}v_{\parallel N} {\rm d}\mu_{\rm N} + \right.$$
$$\left. \frac{Z_{\rm sp}}{T_{\rm R,sp}} \hat{\phi}_{\rm N} \left({\rm e}^{\frac{-\mathcal{E}_{\rm N,sp}}{T_{\rm R,sp}}} \Gamma_0(b_{\rm sp}) - 1 \right) + \frac{\hat{B}_{1\parallel {\rm N}}}{B_{\rm N}} {\rm e}^{\frac{-\mathcal{E}_{\rm N,sp}}{T_{\rm R,sp}}} (\Gamma_0(b_{\rm sp}) - \Gamma_1(b_{\rm sp})) \right] = 0 \tag{1.41}$$

where $\mathcal{E}$ is a combined energy term including the kinetic energy of the toroidal rotation of the plasma and the energy stored in the equilibrium electric field:

$$\mathcal{E} = Ze\langle \Phi_0 \rangle - \frac{1}{2} m \omega_\varphi^2 (R^2 - R_0^2). \tag{1.42}$$

The parallel component of Ampère's law gives

$$\left[k_{\perp {\rm N}}^2 + \beta_{\rm ref} \sum_{\rm sp} \frac{Z_{\rm sp}^2 n_{\rm R,sp}}{m_{\rm R,sp}} {\rm e}^{\frac{-\mathcal{E}_{\rm N,sp}}{T_{\rm R,sp}}} \Gamma_0(b_{\rm sp}) \right] \hat{A}_{1\parallel {\rm N}} =$$
$$2\pi B_{\rm N} \beta_{\rm ref} \sum_{\rm sp} Z_{\rm sp} n_{\rm R,sp} v_{\rm R,sp} \int v_{\parallel N} J_0(k_\perp \rho_{\rm sp}) \hat{g}_{\rm N,sp} {\rm d}v_{\parallel N} {\rm d}\mu_N \tag{1.43}$$

where $\beta_{\rm ref}$ is the reference plasma beta:

$$\beta_{\rm ref} = \frac{2\mu_0 n_{\rm ref} T_{\rm ref}}{B_{\rm ref}^2}. \tag{1.44}$$

The perpendicular component of Ampère's law gives

$$\left[1 + \sum_{\rm sp} \frac{T_{\rm R,sp} n_{\rm R,sp}}{B_{\rm N}^2} \beta_{\rm ref} {\rm e}^{\frac{-\mathcal{E}_{\rm N,sp}}{T_{\rm R,sp}}} (\Gamma_0(b_{\rm sp}) - \Gamma_1(b_{\rm sp})) \right] \hat{B}_{1\parallel {\rm N}} =$$
$$- \sum_{\rm sp} \beta_{\rm ref} \left[2\pi B_{\rm N} T_{\rm R,sp} n_{\rm R,sp} \int \mu_N \hat{J}_1(k_\perp \rho_{\rm sp}) \hat{g}_{\rm N,sp} {\rm d}v_{\parallel N} {\rm d}\mu_{\rm N} \right.$$
$$\left. + {\rm e}^{\frac{-\mathcal{E}_{\rm N,sp}}{T_{\rm R,sp}}} (\Gamma_0(b_{\rm sp}) - \Gamma_1(b_{\rm sp})) \frac{Z_{\rm sp} n_{\rm R,sp}}{2B_{\rm N}} \hat{\phi}_{\rm N} \right]. \tag{1.45}$$

where $\hat{J}_1(z) = \frac{2}{z} J_1(z)$ is a modified first order Bessel function of the first kind.

1.4.2 Field equations in GKW

The gyrokinetic Poisson equation 1.41 and perpendicular Ampère's law 1.45 are coupled through the fluctuating electro-static potential and magnetic compression appearing in both of them. For simpler numerical treatment the two equations have to be decoupled. By introducing the notations

$$\begin{aligned}
I_{1\mathrm{sp}} &= Z_{\mathrm{sp}} n_{\mathrm{R,sp}} 2\pi B_{\mathrm{N}} \int J_0(k_\perp \rho_{\mathrm{sp}}) \hat{g}_{\mathrm{N,sp}} \mathrm{d}v_{\parallel N} \mathrm{d}\mu_{\mathrm{N}} \\
F_{1\mathrm{sp}} &= \frac{Z_{\mathrm{sp}}^2 n_{\mathrm{R,sp}}}{T_{\mathrm{R,sp}}} \left(\mathrm{e}^{\frac{-\mathcal{E}_{\mathrm{N,sp}}}{T_{\mathrm{R,sp}}}} \Gamma_0(b_{\mathrm{sp}}) - 1 \right) \\
B_{1\mathrm{sp}} &= \frac{Z_{\mathrm{sp}} n_{\mathrm{R,sp}}}{B_{\mathrm{N}}} \mathrm{e}^{\frac{-\mathcal{E}_{\mathrm{N,sp}}}{T_{\mathrm{R,sp}}}} \left(\Gamma_0(b_{\mathrm{sp}}) - \Gamma_1(b_{\mathrm{sp}}) \right) \\
I_{2\mathrm{sp}} &= \beta_{\mathrm{ref}} 2\pi B_{\mathrm{N}} T_{\mathrm{R,sp}} n_{\mathrm{R,sp}} \int \mu_N \hat{J}_1(k_\perp \rho_{\mathrm{sp}}) \hat{g}_{\mathrm{N,sp}} \mathrm{d}v_{\parallel N} \mathrm{d}\mu_{\mathrm{N}} \\
F_{2\mathrm{sp}} &= \frac{\beta_{\mathrm{ref}} Z_{\mathrm{sp}} n_{\mathrm{R,sp}}}{2B_{\mathrm{N}}} \mathrm{e}^{\frac{-\mathcal{E}_{\mathrm{N,sp}}}{T_{\mathrm{R,sp}}}} \left(\Gamma_0(b_{\mathrm{sp}}) - \Gamma_1(b_{\mathrm{sp}}) \right) \\
B_{2\mathrm{sp}} &= \frac{T_{\mathrm{R,sp}} n_{\mathrm{R,sp}}}{B_{\mathrm{N}}^2} \beta_{\mathrm{ref}} \mathrm{e}^{\frac{-\mathcal{E}_{\mathrm{N,sp}}}{T_{\mathrm{R,sp}}}} \left(\Gamma_0(b_{\mathrm{sp}}) - \Gamma_1(b_{\mathrm{sp}}) \right)
\end{aligned}$$

equations 1.41 and 1.45 take the form

$$\begin{aligned}
\sum_{\mathrm{sp}} I_{1\mathrm{sp}} + \hat{\phi}_{\mathrm{N}} \sum_{\mathrm{sp}} F_{1\mathrm{sp}} + \hat{B}_{1\parallel\mathrm{N}} \sum_{\mathrm{sp}} B_{1\mathrm{sp}} &= 0 \\
\left(1 + \sum_{\mathrm{sp}} B_{2\mathrm{sp}} \right) \hat{B}_{1\parallel\mathrm{N}} + \sum_{\mathrm{sp}} I_{2\mathrm{sp}} + \hat{\phi}_{\mathrm{N}} \sum_{\mathrm{sp}} F_{2\mathrm{sp}} &= 0.
\end{aligned}$$

Accordingly, the decoupled field equations for $\hat{\Phi}_1$ and $\hat{B}_{1\parallel\mathrm{N}}$ can be written as

$$\begin{aligned}
&\left[\sum_{\mathrm{sp}} F_{1\mathrm{sp}} \left(1 + \sum_{\mathrm{sp}} B_{2\mathrm{sp}} \right) - \sum_{\mathrm{sp}} F_{2\mathrm{sp}} \sum_{\mathrm{sp}} B_{1\mathrm{sp}} \right] \hat{\phi}_{\mathrm{N}} + \\
&\left[\sum_{\mathrm{sp}} I_{1\mathrm{sp}} \left(1 + \sum_{\mathrm{sp}} B_{2\mathrm{sp}} \right) - \sum_{\mathrm{sp}} I_{2\mathrm{sp}} \sum_{\mathrm{sp}} B_{1\mathrm{sp}} \right] = 0
\end{aligned} \tag{1.46}$$

and

$$\begin{aligned}
&\left[\sum_{\mathrm{sp}} F_{1\mathrm{sp}} \left(1 + \sum_{\mathrm{sp}} B_{2\mathrm{sp}} \right) - \sum_{\mathrm{sp}} F_{2\mathrm{sp}} \sum_{\mathrm{sp}} B_{1\mathrm{sp}} \right] \hat{B}_{1\parallel\mathrm{N}} + \\
&\left[\sum_{\mathrm{sp}} I_{2\mathrm{sp}} \sum_{\mathrm{sp}} F_{1\mathrm{sp}} - \sum_{\mathrm{sp}} I_{1\mathrm{sp}} \sum_{\mathrm{sp}} F_{2\mathrm{sp}} \right] = 0.
\end{aligned} \tag{1.47}$$

The implementation of the field equations in GKW follows the above notation.

Chapter 2

Gyrokinetic practice

2.1 Normalisations

Of course, all quantities in the code are normalised. Below, this normalisation is discussed in detail. The velocity will be normalised by the thermal velocity of each of the species. This has the advantage that the grid in velocity space can always be defined relative to the thermal velocity. Quantities like the potential, however, must be normalised by a reference temperature since it is species independent. A similar argument applies to the vector potential and the time.

We define a reference mass $m_{\rm ref}$, a reference thermal velocity $v_{\rm thref}$, a reference density $n_{\rm ref}$, a reference temperature $T_{\rm ref}$, a reference magnetic field $B_{\rm ref}$ evaluated on the magnetic axis, and a reference major radius $R_{\rm ref}$. These are not unrelated since

$$T_{\rm ref} = \frac{1}{2} m_{\rm ref} v_{\rm thref}^2, \qquad \rho_{\rm ref} = \frac{m_{\rm ref} v_{\rm thref}}{e B_{\rm ref}}. \tag{2.1}$$

Furthermore we define a normalised Larmor radius to be

$$\rho_* = \rho_{\rm ref}/R_{\rm ref}. \tag{2.2}$$

The reference values are used to define, for each species, a dimensionless mass m_R, a dimensionless thermal velocity v_R, a dimensionless density n_R, a dimensionless temperature T_R, and a dimensionless centrifugal energy $\mathcal{E}_R$

$$m_R = \frac{m}{m_{\rm ref}}, \qquad v_R = \frac{v_{\rm th}}{v_{\rm thref}}, \qquad n_R = \frac{n_{R_0}}{n_{\rm ref}}, \qquad T_R = \frac{T}{T_{\rm ref}}, \qquad \mathcal{E}_R = \frac{\mathcal{E}_\Omega}{T_R T_{\rm ref}}. \tag{2.3}$$

The fields are made dimensionless using only reference values

$$\phi = \rho_* \frac{T_{\rm ref}}{e} \phi_N, \qquad A_\parallel = B_{\rm ref} R_{\rm ref} \rho_*^2 A_{\parallel N}, \qquad B_{1\parallel} = B_{\rm ref} \rho_* B_{1\parallel N} \qquad \chi = \rho_* \frac{T_{\rm ref}}{e} \chi_N, \qquad \Phi = \frac{T_{\rm ref}}{e} \Phi_N. \tag{2.4}$$

Here the index N refers to a normalised quantity. Note that factors of ρ_* have been added in the definitions of the normalised perturbed fields. These factors are chosen such that the normalised quantities are of the order 1.

Also time t, magnetic field B, angular rotation frequency Ω, and the major radius R, are made dimensionless using the reference values

$$\begin{aligned} t &= R_{\rm ref} t_N / v_{\rm thref}, \qquad & B &= B_{\rm ref} B_N, \\ \Omega &= v_{\rm thref} \Omega_N / R_{\rm ref}, \qquad & R &= R_{\rm ref} R_N. \end{aligned} \tag{2.5}$$

The velocity space coordinates, however, are made dimensionless with the help of the thermal velocity of the species

$$v_\parallel = v_{\parallel N} v_{\rm th}, \qquad \mu = \frac{m v_{\rm th}^2}{B_{\rm ref}} \mu_N. \tag{2.6}$$

It is then convenient to normalise the distribution functions according to

$$f = \rho_* \frac{n_{R_0}}{v_{\rm th}^3} f_N, \qquad F_M = \frac{n_{R_0}}{v_{\rm th}^3} F_{MN}. \tag{2.7}$$

The gradient of density and temperature are made dimensionless using the density and temperature of the species under consideration, but the plasma rotation is largely a bulk motion of all the species together and its gradient is normalised using the reference thermal velocity $v_{\rm thref}$

$$\frac{1}{L_{n,N}} = \frac{R_{\rm ref}}{L_n} = -\frac{1}{n_{R_0}} \frac{\partial n_{R_0}}{\partial \psi}, \qquad \frac{1}{L_{T,N}} = \frac{R_{\rm ref}}{L_T} = -\frac{1}{T} \frac{\partial T}{\partial \psi}, \qquad u'_N = -\frac{R_{\rm ref}}{v_{\rm thref}} \frac{\partial \omega_\phi}{\partial \psi}. \tag{2.8}$$

The gradient length scales are normalised with $R_{\rm ref}$, but note that the same quantities are often written as R/L_n and R/L_T elsewhere in the literature. Note also that the radial coordinate ψ is normalised:

$$\psi = \frac{R_{\rm max} - R_{\rm min}}{2R_{\rm ref}}, \tag{2.9}$$

where $R_{\rm max}$ ($R_{\rm min}$) is the maximum (minimum) major radius of the flux surface. For circular surfaces, for instance, $\psi = r/R_{\rm ref} = \epsilon$, where r is the minor radius. For the $s-\alpha$ geometry, you can take $R_{\rm ref}$ to have any value you like, but it is simplest to always use the magnetic axis. For the Circ, Miller and Chease $R_{\rm ref}$ has a specific meaning defined by the geometry. All gradients are normalised using the major radius $R_{\rm ref}$, i.e.

$$\nabla = \frac{1}{R_{\rm ref}} \nabla_N. \tag{2.10}$$

The poloidal flux Ψ used in the derivation of the metric tensors is normalised according to

$$\Psi = R_{\rm ref}^2 B_{\rm ref} \Psi_N. \tag{2.11}$$

The strength of the electromagnetic effects is determined by the plasma beta (β). Although β is dimensionless we define a different dimensionless beta (β_N) for the use in the code. This β_N is directly related to the reference values

$$\beta_N = \frac{n_{\rm ref} T_{\rm ref}}{B_{\rm ref}^2 / 2\mu_0}. \tag{2.12}$$

Finally, the wave vectors introduced in the spectral representation arise from the perpendicular gradient of a fluctuating quantity and will therefore be normalised to $\rho_{\rm ref}$:

$$k = \frac{k_N}{\rho_{\rm ref}}. \tag{2.13}$$

In all the following sections we will, of course, use only the normalised quantities. The index N is then dropped for convenience.

2.2 Geometry

GKW is formulated in field aligned Hamada coordinates [36], i.e. the contravariant components of both the poloidal B^s and toroidal magnetic field B^γ are flux functions. The safety factor is then determined by the ratio $q = B^\gamma / B^s$. Starting from an orthogonal coordinate system (ψ, θ, φ), where ψ is the radial coordinate (i.e. $\mathbf{B}\cdot\nabla\psi = 0$), θ is the poloidal angle (upward on the outboard midplane), and φ is the toroidal angle (clockwise when viewed from above), and using the transformations

$$s = s(\psi,\theta), \qquad \gamma = \gamma(\psi,\theta,\varphi), \tag{2.14}$$

one can derive [40, 41] (see Appendix A.1 for more details):

$$s(\theta,\psi) = \int_0^\theta \frac{\mathrm{d}\theta'}{\mathbf{B}\cdot\nabla\theta'} \bigg/ \oint \frac{\mathrm{d}\theta'}{\mathbf{B}\cdot\nabla\theta'}, \tag{2.15}$$

$$\gamma = \frac{\varphi}{2\pi} + s_\mathrm{B}\frac{RB_t}{2\pi}\int_0^\theta \frac{\mathrm{d}\theta'}{\mathbf{B}\cdot\nabla\theta'}\left[\left\{\frac{1}{R^2}\right\} - \frac{1}{R^2}\right], \tag{2.16}$$

with

$$B^s = 1\bigg/\oint \frac{\mathrm{d}\theta'}{\mathbf{B}\cdot\nabla\theta'} \qquad B^\gamma = s_\mathrm{B}\frac{RB_t}{2\pi}\left\{\frac{1}{R^2}\right\}. \tag{2.17}$$

In the equations above, the magnetic field is decomposed as

$$\mathbf{B} = s_\mathrm{B} RB_t \nabla\varphi + s_\mathrm{j}\nabla\varphi\times\nabla\Psi, \tag{2.18}$$

where $B_t > 0$ is the toroidal component of the magnetic field, $s_\mathrm{B} = \pm 1$ and $s_\mathrm{j} = \pm 1$ the sign of the magnetic field and plasma current (positive in the direction of $\nabla\varphi$) and Ψ the normalised poloidal flux ($\nabla\Psi$ points from the magnetic axis to the plasma edge). The brackets $\{\}$ denote the flux surface average, which in the transformed coordinates can be written (more details in A.6) as

$$\{g\} = \oint g\,\mathrm{d}s. \tag{2.19}$$

The normalising constants have been chosen such that the domain [-1/2,1/2] in s corresponds to one poloidal turn and the domain [-1/2,1/2] in γ corresponds to one toroidal turn. Note that the transformations of Eq. (2.14) leave the angle γ to be an ignorable coordinate, since any quantity that is independent of φ will also be independent of γ after the transformation.

The coordinates given above are transformed using a simple linear transformation

$$\zeta = qs - \gamma \qquad \text{where} \qquad q = \frac{B^\gamma}{B^s} \tag{2.20}$$

to make them field aligned, i.e. $\mathbf{B}\cdot\nabla = B^s\partial/\partial s$ as $B^\zeta = 0$ and $B^\psi = 0$. Note that the coordinate transformation above flips the sign of the toroidal angle. The right handed coordinate system can therefore be defined as (ψ, ζ, s). The Jacobian of the new coordinate system can be expressed in terms of the original Jacobian through

$$J_{\psi\zeta s} = 2\pi\mathbf{B}\cdot\nabla\theta\oint\frac{\mathrm{d}\theta'}{\mathbf{B}\cdot\nabla\theta'}J_{\psi\theta\varphi}. \tag{2.21}$$

We note here that the coordinates ψ, s and ζ are chosen dimensionless with ψ being a normalised 'minor radius' of the flux surface, see Eq. (2.9). We shall refer to ζ as the 'bi-normal' coordinate since it is neither strictly toroidal or poloidal.

Close inspection of the gyrokinetic equation (1.31) shows that it contains parallel derivatives

$$\mathbf{b}\cdot\nabla = \mathcal{F}\frac{\partial}{\partial s} \quad\to\quad \mathcal{F} = \frac{B^s}{B}, \tag{2.22}$$

and perpendicular drifts, with the latter all involving a cross product with the magnetic field. It is then convenient to define the tensor

$$\mathcal{E}^{\alpha\beta} = \frac{1}{2B}(\nabla x_\alpha \times \nabla x_\beta)\cdot\mathbf{b}. \tag{2.23}$$

The convection connected with the velocity $\mathbf{v}_\chi$ can be directly expressed in this tensor times the derivatives of the potential towards x_α and the distribution towards x_β. It is worth noting that the magnetic field can be expressed as

$$\mathbf{B} = 2\pi s_{\rm j} \nabla\Psi \times \nabla\zeta, \tag{2.24}$$

which immediately leads to

$$\mathcal{E}^{\psi\zeta} = \frac{s_{\rm j}}{4\pi}\frac{\partial\psi}{\partial\Psi}, \tag{2.25}$$

showing that $\mathcal{E}^{\psi\zeta}$ is constant on a flux surface. For convenience we define two more quantities

$$\mathcal{D}^\alpha = -2\mathcal{E}^{\alpha\beta}\frac{1}{B}\frac{\partial B}{\partial x_\beta}, \qquad \mathcal{G} = \mathcal{F}\frac{\partial \ln B}{\partial s}, \tag{2.26}$$

with $\mathcal{D}$ related to the grad B drift, and $\mathcal{G}$ to the trapping. Finally, the Coriolis drift and centrifugal drift will enter the equations through the one forms $\mathcal{H}$ and $\mathcal{I}$ respectively

$$\mathcal{H}^\alpha = -\frac{s_{\rm B}}{B\Omega}\mathbf{\Omega}_\perp \cdot \nabla x_\alpha, \qquad \mathcal{I}^\alpha = \frac{s_{\rm B}}{2B}(\nabla x_\alpha \times \nabla R^2)\cdot \mathbf{b} \tag{2.27}$$

and the centrifugal potential is calculated (see Sec. 4.1) using the quantities

$$\mathcal{J} = R^2 - R_0^2, \qquad \mathcal{K} = \frac{\partial\mathcal{J}}{\partial\psi}\bigg|_s, \qquad \mathcal{L} = \frac{\partial(R_0^2)}{\partial\psi}\bigg|_s. \tag{2.28}$$

For transformations of the shearing rate we also define

$$\mathcal{M} = \frac{\partial^2\Psi}{\partial\psi^2} = \frac{s_{\rm j}}{4\pi}\frac{\partial}{\partial\psi}\left(\frac{1}{\mathcal{E}^{\psi\zeta}}\right). \tag{2.29}$$

The tensor elements defined above all have similar magnitude. It must be realised though that these elements are multiplied with the derivatives towards the respective coordinate. Since the derivatives along the field are much smaller than the derivatives in the perpendicular plane, the tensor elements containing the coordinate s are usually neglected (i.e. $\mathcal{D}^s = \mathcal{E}^{s\psi} = \mathcal{E}^{s\zeta} = \mathcal{H}^s = 0$), except in the case of those multiplied with the background potential Φ, or when higher ρ^* effects are included.

GKW uses the local limit. Central to this approximation is the small scales of the turbulent fluctuations. All background plasma parameters are assumed homogeneous across the simulation domain perpendicular to the field, i.e. there is no dependence of the parameters on the coordinates ζ and ψ. The turbulence is homogeneous in the plane perpendicular to the magnetic field and periodic boundary conditions apply. Note that the dependence of equilibrium quantities on the 'parallel' coordinate s are kept. All the tensors $\mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H}, \mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{M}$ as well as the metric tensor, are therefore, functions of the parallel coordinate s only.

2.2.1 s-α

The simplified '$s - \alpha$' equilibrium (with $\alpha = 0$) is directly implemented in the code through pre-defined tensor elements. The latter model equilibrium is the simplest choice of the geometry of a tokamak with the flux surfaces being circular and having a small inverse aspect ratio $\epsilon = r/R \ll 1$, with r being the minor radius of the surface and $R = R_{\rm ref}$. Only the lowest order in an ϵ expansion is kept in the definition of all the tensors. The coordinates can then be approximated as

$$\psi = \epsilon = \frac{r}{R}, \qquad \zeta = \frac{s_{\rm B}s_{\rm j}}{2\pi}[|q|\theta - \varphi], \qquad s = \frac{\theta}{2\pi}, \tag{2.30}$$

with the (normalised) gradients

$$\nabla\psi = \mathbf{e}_r, \qquad \nabla\zeta = \frac{s_{\rm B}s_{\rm j}|q|}{2\pi\epsilon}[\mathbf{e}_\theta + \hat{s}\theta\mathbf{e}_r], \qquad \nabla s = \frac{1}{2\pi\epsilon}\mathbf{e}_\theta, \tag{2.31}$$

where $\hat{s} = (r/q)\mathrm{d}q/\mathrm{d}r$ is the magnetic shear. Note that the gradient of the toroidal angle is ordered small compared with the gradient of the poloidal angle (times q) and is neglected in the gradient of ζ. The contravariant components of the magnetic field are

$$B^s = s_\mathrm{j}\frac{B_p}{2\pi r} \qquad B^\zeta = -s_\mathrm{B}\frac{B_t}{2\pi R}, \tag{2.32}$$

where B_p is the poloidal magnetic field. The gradient of the magnetic field strength is assumed to be in the direction of ∇R and the drift term is approximated as

$$\mathbf{B} \times \nabla B = -s_\mathrm{B} B^2[\cos\theta \mathbf{e}_\theta + \sin\theta \mathbf{e}_r]. \tag{2.33}$$

Using these expressions one can evaluate the tensors to be

$$\mathcal{F} = \frac{s_\mathrm{j}}{2\pi|q|}, \qquad \mathcal{D}^\psi = -s_\mathrm{B}\sin(2\pi s), \qquad \mathcal{D}^\zeta = -\frac{s_\mathrm{j}|q|}{2\pi\epsilon}[\cos(2\pi s) + 2\pi\hat{s}s\sin(2\pi s)], \tag{2.34}$$

$$\mathcal{I}^\psi = \mathcal{H}^\psi = \mathcal{D}^\psi, \qquad \mathcal{I}^\zeta = \mathcal{H}^\zeta = \mathcal{D}^\zeta, \qquad \mathcal{G} = \frac{s_\mathrm{j}\epsilon}{|q|}\sin(2\pi s), \tag{2.35}$$

$$\mathcal{E}^{\psi\zeta} = s_\mathrm{j}\frac{|q|}{4\pi\epsilon} = -\mathcal{E}^{\zeta\psi}, \qquad \mathcal{E}^{\psi s} = \frac{s_\mathrm{B}}{4\pi\epsilon}, \qquad \mathcal{E}^{\zeta s} = \frac{s_\mathrm{j}|q|}{4\pi\epsilon^2}\hat{s}s, \qquad \mathcal{M} = \frac{1}{q} - \frac{\hat{s}}{q} \tag{2.36}$$

$$\mathcal{J} = 2\epsilon\cos(2\pi s), \qquad \mathcal{K} = 2\cos(2\pi s) - \mathcal{L}. \tag{2.37}$$

where the two choices for R_0 (mag. axis R and $R(s = 0)$, see 4.1) give

$$R_0 = 1 \Rightarrow \mathcal{L} = 0, \qquad R_0 = 1 + \epsilon \Rightarrow \mathcal{L} = 2. \tag{2.38}$$

respectively.

Finally the metric elements are

$$g^{\zeta\zeta} = \left(\frac{q}{2\pi\epsilon}\right)^2 + \left(\frac{q}{\epsilon}\hat{s}s\right)^2, \qquad g^{\zeta\psi} = \frac{s_\mathrm{B}s_\mathrm{j}|q|}{\epsilon}\hat{s}s, \qquad g^{\psi\psi} = 1. \tag{2.39}$$

All numerical results presented in this manual have been obtained with the geometry coefficients given above, except the benchmark of the full geometry and Miller geometry treatment.

2.2.2 Circular geometry

The $s - \alpha$ model has been shown to be inconsistent in the ϵ ordering of the different terms involved [44] and for some nonlinear runs can also be numerically unstable for the zonal mode. In GKW the improved ad-hoc circular equilibrium model of Ref. [44] is also implemented.

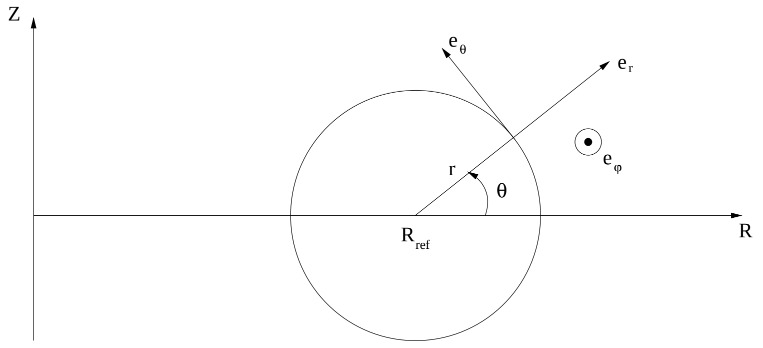


Figure 2.1: Circular flux surface centred on $R = R_\mathrm{ref}$, $Z = 0$ and (r, θ, φ) coordinates.

In this section, unless explicitly denoted by a N subscript, the various quantities are not normalised. The starting assumption of the model is that the flux surfaces are circular and concentric with the poloidal flux being a function of the radial coordinate only $\Psi = \Psi(r)$, see figure 2.1. This is a valid assumption in the case of small ϵ and small β. It simply means that the terms of order ϵ^2 are neglected and that the Shafranov shift is considered to be of order ϵ^2. For simplicity, the various terms are not expanded in ϵ in the following calculation, but one has to remember that the results are only valid up to order ϵ. The second assumption of the model is that the radial derivative of the poloidal flux is given by:

$$\frac{\partial \Psi}{\partial r} = \frac{r B_{\text{ref}}}{\bar{q}} \tag{2.40}$$

where B_{ref} is the value of the magnetic field at R_{ref} taken to be the major radius of the magnetic axis (i.e. the centre of the flux surfaces) and $\bar{q} > 0$ is a parameter directly related to the safety factor q. The flux surfaces are parametrised by $R = R_{\text{ref}} + r\cos\theta$ and $Z = r\sin\theta$ in the (r, θ, φ) coordinates system. The inverse aspect ratio is defined as $\epsilon = r/R_{\text{ref}}$ and it might be worth reminding that:

$$\nabla\epsilon = \frac{1}{R_{\text{ref}}}\mathbf{e_r}, \qquad \nabla\theta = \frac{1}{\epsilon R_{\text{ref}}}\mathbf{e}_\theta, \qquad \nabla\varphi = \frac{1}{R}\mathbf{e}_\varphi \tag{2.41}$$

Another useful quantity is the Jacobian of the (Ψ, θ, φ) coordinates system which can be expressed as

$$J_{\Psi\theta\varphi}{}^{-1} = \nabla\Psi \times \nabla\theta \cdot \nabla\varphi = \frac{\partial \Psi}{\partial r} J_{r\theta\varphi}{}^{-1} = \frac{\partial \Psi}{\partial r}\frac{1}{rR} = \frac{B_{\text{ref}}}{\bar{q}R} \tag{2.42}$$

As usual, the magnetic field is decomposed into its toroidal and poloidal components:

$$\mathbf{B} = s_{\text{B}} F(\Psi)\nabla\varphi + s_{\text{j}}\nabla\varphi \times \nabla\Psi \tag{2.43}$$

One furthermore assumes that $F(\Psi) = R_{\text{ref}}B_{\text{ref}}$ and the magnetic field can then be written as

$$\mathbf{B} = R_{\text{ref}}B_{\text{ref}}\left[s_{\text{B}}\nabla\varphi + s_{\text{j}}\frac{1}{\bar{q}}\frac{\epsilon^2}{1+\epsilon\cos\theta}\nabla\theta\right] \tag{2.44}$$

The definition of the safety factor is

$$q = \frac{1}{2\pi}\int_0^{2\pi}\frac{\mathbf{B}\cdot\nabla\varphi}{\mathbf{B}\cdot\nabla\theta}\,\mathrm{d}\theta \tag{2.45}$$

which in the model considered here leads to

$$q = \frac{1}{2\pi}\int_0^{2\pi} s_{\text{B}}s_{\text{j}}J_{\Psi\theta\varphi}\frac{R_{\text{ref}}B_{\text{ref}}}{R^2}\,\mathrm{d}\theta = s_{\text{B}}s_{\text{j}}\frac{\bar{q}}{2\pi}\int_0^{2\pi}\frac{1}{1+\epsilon\cos\theta}\,\mathrm{d}\theta = s_{\text{B}}s_{\text{j}}\frac{\bar{q}}{\sqrt{1-\epsilon^2}} \tag{2.46}$$

which makes explicit the relationship between q and $\bar{q}$.

We now have all the ingredients to derive the field aligned coordinates (ψ, ζ, s). In GKW, ψ is defined to be ϵ and we will use this notation in the following to avoid possible confusion with the poloidal flux Ψ. The definition of the parallel coordinate s is

$$s = \int_0^{\theta}\frac{\mathrm{d}\theta}{\mathbf{B}\cdot\nabla\theta} \Big/ \int_0^{2\pi}\frac{\mathrm{d}\theta}{\mathbf{B}\cdot\nabla\theta} \tag{2.47}$$

So, we first calculate

$$\int_0^{\theta}\frac{\mathrm{d}\theta}{\mathbf{B}\cdot\nabla\theta} = \int_0^{\theta} s_{\text{j}}J_{\Psi\theta\varphi}\,\mathrm{d}\theta = s_{\text{j}}\bar{q}\frac{R_{\text{ref}}}{B_{\text{ref}}}\int_0^{\theta}[1+\epsilon\cos\theta]\,\mathrm{d}\theta = s_{\text{j}}\bar{q}\frac{R_{\text{ref}}}{B_{\text{ref}}}[\theta + \epsilon\sin\theta], \tag{2.48}$$

to get

$$s = \frac{1}{2\pi}[\theta + \epsilon\sin\theta] \tag{2.49}$$

and

$$\frac{\partial s}{\partial \epsilon} = \frac{1}{2\pi}\sin\theta, \tag{2.50}$$

$$\frac{\partial s}{\partial \theta} = \frac{1}{2\pi}[1+\epsilon\cos\theta], \tag{2.51}$$

$$\frac{\partial s}{\partial \varphi} = 0. \tag{2.52}$$

The definition of the ζ coordinate is:

$$\zeta = -\frac{\varphi}{2\pi} + s_{\mathrm{B}}s_{\mathrm{j}}\frac{F}{2\pi}\int_0^\theta \frac{J_{\Psi\theta\varphi}}{R^2}\,\mathrm{d}\theta \tag{2.53}$$

which in the ad-hoc circular model gives

$$\zeta = -\frac{\varphi}{2\pi} + s_{\mathrm{B}}s_{\mathrm{j}}\frac{\bar{q}}{2\pi}\int_0^\theta \frac{\mathrm{d}\theta}{1+\epsilon\cos\theta} \tag{2.54}$$

Noticing that

$$\int_0^\theta \frac{\mathrm{d}\theta}{1+\epsilon\cos\theta} = \frac{2}{\sqrt{1-\epsilon^2}}\arctan\left[\sqrt{\frac{1-\epsilon}{1+\epsilon}}\tan\frac{\theta}{2}\right], \tag{2.55}$$

one gets

$$\zeta = -\frac{\varphi}{2\pi} + s_{\mathrm{B}}s_{\mathrm{j}}\frac{|q|}{\pi}\arctan\left[\sqrt{\frac{1-\epsilon}{1+\epsilon}}\tan\frac{\theta}{2}\right] \tag{2.56}$$

and

$$\frac{\partial \zeta}{\partial \epsilon} = s_{\mathrm{B}}s_{\mathrm{j}}\frac{|q|}{\epsilon\pi}\left[\hat{s}\arctan\left[\sqrt{\frac{1-\epsilon}{1+\epsilon}}\tan\frac{\theta}{2}\right] - \frac{\epsilon\tan\frac{\theta}{2}}{\sqrt{1-\epsilon^2}\left(1+\epsilon+(1-\epsilon)\tan^2\frac{\theta}{2}\right)}\right], \tag{2.57}$$

$$\frac{\partial \zeta}{\partial \theta} = s_{\mathrm{B}}s_{\mathrm{j}}\frac{|q|}{2\pi}\frac{\sqrt{1-\epsilon^2}}{1+\epsilon\cos\theta}, \tag{2.58}$$

$$\frac{\partial \zeta}{\partial \varphi} = -\frac{1}{2\pi}. \tag{2.59}$$

From the decomposition of ∇s and $\nabla\zeta$ as a function of $\nabla\epsilon$, $\nabla\theta$ and $\nabla\varphi$

$$\nabla s = \frac{\partial s}{\partial \epsilon}\nabla\epsilon + \frac{\partial s}{\partial \theta}\nabla\theta + \frac{\partial s}{\partial \varphi}\nabla\varphi \tag{2.60}$$

$$\nabla \zeta = \frac{\partial \zeta}{\partial \epsilon}\nabla\epsilon + \frac{\partial \zeta}{\partial \theta}\nabla\theta + \frac{\partial \zeta}{\partial \varphi}\nabla\varphi, \tag{2.61}$$

the calculation of the metric tensor is straightforward:

$$g_N^{\epsilon\epsilon} = R_{\mathrm{ref}}^2 g^{\epsilon\epsilon} = 1, \tag{2.62}$$

$$g_N^{\epsilon\zeta} = R_{\mathrm{ref}}^2 g^{\epsilon\zeta} = \frac{\partial\zeta}{\partial\epsilon}, \tag{2.63}$$

$$g_N^{\epsilon s} = R_{\mathrm{ref}}^2 g^{\epsilon s} = \frac{1}{2\pi}\sin\theta, \tag{2.64}$$

$$g_N^{\zeta\zeta} = R_{\mathrm{ref}}^2 g^{\zeta\zeta} = \left[\frac{\partial\zeta}{\partial\epsilon}\right]^2 + \left[\frac{1}{2\pi}\frac{1}{1+\epsilon\cos\theta}\right]^2\left[1+\frac{q^2}{\epsilon^2}(1-\epsilon^2)\right] \tag{2.65}$$

$$g_N^{\zeta s} = R_{\mathrm{ref}}^2 g^{\zeta s} = \frac{\sin\theta}{2\pi}\frac{\partial\zeta}{\partial\epsilon} + s_{\mathrm{B}}s_{\mathrm{j}}\frac{|q|}{(2\pi\epsilon)^2}\sqrt{1-\epsilon^2} \tag{2.66}$$

$$g_N^{ss} = R_{\mathrm{ref}}^2 g^{ss} = \frac{1}{4\pi^2}\left[1+\frac{2}{\epsilon}\cos\theta+\frac{1}{\epsilon^2}\right]. \tag{2.67}$$

To calculate the other metric elements one needs the norm of the magnetic field

$$B = \sqrt{\mathbf{B}\cdot\mathbf{B}} = \frac{B_{\mathrm{ref}}}{1+\epsilon\cos\theta}\left[1+\frac{1}{q^2}\frac{\epsilon^2}{1-\epsilon^2}\right]^{1/2} \tag{2.68}$$

and its gradient

$$\nabla B = \frac{\partial B}{\partial \epsilon}\nabla\epsilon + \frac{\partial B}{\partial\theta}\nabla\theta \tag{2.69}$$

which involves

$$\frac{\partial B}{\partial\epsilon} = \left[-\frac{\cos\theta}{1+\epsilon\cos\theta} + \frac{\epsilon}{\epsilon^2+q^2(1-\epsilon^2)}(1-\hat{s}+\frac{\epsilon^2}{1-\epsilon^2})\right]B \tag{2.70}$$

$$\frac{\partial B}{\partial\theta} = \frac{\epsilon\sin\theta}{1+\epsilon\cos\theta}B \tag{2.71}$$

From the definition

$$\mathcal{D}_N^\alpha = \frac{R_{\mathrm{ref}}^2}{B^2 B_N}\mathbf{B}\times\nabla B\cdot\nabla x^\alpha \tag{2.72}$$

and some straightforward algebra (which has to be understood as “I am by far too lazy to write down all the steps in LaTeX”) we get the components of the tensor related to the curvature and ∇B drift:

$$\mathcal{D}_N^\epsilon = -s_{\mathrm{B}}\sin\theta\frac{1}{1+\frac{1}{q^2}\frac{\epsilon^2}{1-\epsilon^2}} \tag{2.73}$$

$$\mathcal{D}_N^\zeta = \left[\frac{s_{\mathrm{j}}}{2\pi}\frac{1}{B}\frac{\partial B}{\partial\epsilon}\left(\frac{|q|\sqrt{1-\epsilon^2}}{\epsilon}+\frac{\epsilon}{|q|\sqrt{1-\epsilon^2}}\right)-s_{\mathrm{B}}\frac{\partial\zeta}{\partial\epsilon}\sin\theta\right]\frac{1}{1+\frac{1}{q^2}\frac{\epsilon^2}{1-\epsilon^2}} \tag{2.74}$$

$$\mathcal{D}_N^s = \frac{s_{\mathrm{B}}}{2\pi}\left[\frac{1}{\epsilon B}\frac{\partial B}{\partial\epsilon}(1+\epsilon\cos\theta)^2-\sin^2\theta\right]\frac{1}{1+\frac{1}{q^2}\frac{\epsilon^2}{1-\epsilon^2}} \tag{2.75}$$

(2.76)

The tensor related to the $\mathbf{E}\times\mathbf{B}$ drift can be obtained from the metric tensor elements:

$$\mathcal{E}_N^{\alpha\beta} = s_{\mathrm{j}}\frac{\pi R_{\mathrm{ref}}^2}{BB_N}\frac{\partial\Psi}{\partial\epsilon}(g^{\alpha\epsilon}g^{\beta\zeta}-g^{\alpha\zeta}g^{\beta\epsilon}) \tag{2.77}$$

which for the ad-hoc model gives:

$$\mathcal{E}_N^{\alpha\beta} = s_{\mathrm{j}}\frac{\pi}{B_N^2}\frac{\epsilon}{|q|\sqrt{1-\epsilon^2}}(g_N^{\alpha\epsilon}g_N^{\beta\zeta}-g_N^{\alpha\zeta}g_N^{\beta\epsilon}) \tag{2.78}$$

We the calculate the tensor related to the parallel derivative

$$\mathcal{F}_N = R_{\mathrm{ref}}\frac{\mathbf{B}\cdot\nabla s}{B} = \frac{s_{\mathrm{j}}}{2\pi|q|B_N}\frac{1}{\sqrt{1-\epsilon^2}} \tag{2.79}$$

and the tensor related to the trapping

$$\mathcal{G}_N = \frac{R_{\mathrm{ref}}\mathbf{B}\cdot\nabla B}{B^2} = s_{\mathrm{j}}\frac{\epsilon B_N\sin\theta}{|q|\sqrt{1-\epsilon^2}}\frac{1}{1+\frac{1}{q^2}\frac{\epsilon^2}{1-\epsilon^2}}. \tag{2.80}$$

For the tensor related to the Coriolis drift

$$\mathcal{H}_N^\alpha = \frac{R_{\mathrm{ref}}}{B_N\Omega}\mathbf{\Omega}_\perp\cdot\nabla x^\alpha, \tag{2.81}$$

we first decompose $\mathbf{\Omega}$ as

$$\mathbf{\Omega} = -s_{\mathrm{B}}\Omega\nabla Z = -s_{\mathrm{B}}R_{\mathrm{ref}}\Omega\left[\sin\theta\nabla\epsilon+\epsilon\cos\theta\nabla\theta\right] \tag{2.82}$$

to get

$$\mathbf{\Omega} \cdot \mathbf{b} = -s_{\rm B} s_{\rm j} \Omega \frac{\epsilon \cos\theta}{|q|\sqrt{1-\epsilon^2}} \frac{R_{\rm ref} B_{\rm ref}}{RB} \tag{2.83}$$

and finally obtain

$$\mathcal{H}_N^\epsilon = -\frac{s_{\rm B}}{B_N} \sin\theta \tag{2.84}$$

$$\mathcal{H}_N^\zeta = -\frac{s_{\rm B}}{B_N} \sin\theta \frac{\partial\zeta}{\partial\epsilon} - \frac{s_{\rm j}|q|\cos\theta}{2\pi\epsilon B_N} \frac{\sqrt{1-\epsilon^2}}{1+\epsilon\cos\theta} \tag{2.85}$$

$$\mathcal{H}_N^s = -\frac{s_{\rm B}}{2\pi B_N} \sin^2\theta + \frac{s_{\rm B}}{2\pi B_N}(1+\epsilon\cos\theta)\left[-\frac{\cos\theta}{\epsilon} + \frac{\epsilon\cos\theta}{q^2(1-\epsilon^2)} \frac{1}{1+\frac{1}{q^2}\frac{\epsilon^2}{1-\epsilon^2}}\right]. \tag{2.86}$$

Using

$$\nabla R = R_{\rm ref}\left[\cos\theta\nabla\epsilon - \epsilon\sin\theta\nabla\theta\right] \tag{2.87}$$

the tensor related to the centrifugal drift, can be expressed as

$$\mathcal{I}_N^\alpha = s_{\rm j} \frac{2\pi}{B_N^2} \frac{\epsilon(1+\epsilon\cos\theta)}{|q|\sqrt{1-\epsilon^2}} \left[g_N^{\epsilon\alpha} \left[\cos\theta \frac{\partial\zeta}{\partial\epsilon} - s_{\rm B} s_{\rm j} \frac{|q|}{2\pi\epsilon} \frac{\sin\theta\sqrt{1-\epsilon^2}}{1+\epsilon\cos\theta}\right] - g_N^{\zeta\alpha}\cos\theta \right] \tag{2.88}$$

We also have

$$\mathcal{M} = \frac{1}{q\sqrt{1-\epsilon^2}}\left(1 - \frac{\epsilon}{1-\epsilon^2} - \frac{\hat{s}}{q}\right). \tag{2.89}$$

Finally, the elements used to calculate the centrifugal potential are

$$\mathcal{J}_N = (1+\epsilon\cos\theta)^2 - \frac{R_0^2}{R_{\rm ref}^2} \qquad \mathcal{K}_N = \left.\frac{\partial\mathcal{J}_N}{\partial\epsilon}\right|_s = 2(\cos\theta+\epsilon) - \mathcal{L}_N \tag{2.90}$$

where the transformation between partial deriviatives at constant θ and s gives

$$\frac{1}{R_{\rm ref}} \left.\frac{\partial R}{\partial\epsilon}\right|_s = \cos(\theta) + \frac{\epsilon\sin^2(\theta)}{1+\epsilon\cos(\theta)} \tag{2.91}$$

which has been used to calculate $\mathcal{K}$. The two choices for R_0 (magnetic axis value or $R(s=0)$) give

$$R_0 = R_{\rm ref} \Rightarrow \mathcal{L}_N = 0, \quad R_0 = (1+\epsilon)R_{\rm ref} \Rightarrow \mathcal{L}_N = 2(1+\epsilon) \tag{2.92}$$

respectively.

2.2.3 Miller geometry

Miller geometry is a good compromise between the flexibility of the circ geometry and the precision of a global equilibrium. In this section the Miller parametrisation first done in [37] is described. Then the derivation of the metric tensor in GKW coordinate system (ψ, ζ, s) and the derivatives of the magnetic field are shown using the implementation method of [38].
To avoid any confusion between the poloidal magnetic flux and the radial coordinate the latter will be called r through the entire section. For the miller geometry, $R_{\rm ref} = (R_{\rm max} + R_{\rm min})/2$ which is the geometric axis of the flux surface in question.

Parametrisation (Input List)

The following parameters are used to describe a magnetic flux surface in (R,Z) plane : κ (elongation), δ (triangularity), ζ (squareness), $R_{\rm mil}$, $Z_{\rm mil}$, and their radial derivatives s_κ, s_δ(definition of[37]), s_ζ, $\frac{{\rm d}R_{\rm mil}}{{\rm d}r}$, $\frac{{\rm d}Z_{\rm mil}}{{\rm d}r}$ defined such that:

$$s_\kappa = \frac{r}{\kappa}\frac{{\rm d}\kappa}{{\rm d}r}, \qquad s_\delta = r\frac{\frac{{\rm d}\delta}{{\rm d}r}}{\sqrt{1-\delta^2}}, \qquad s_\zeta = r\frac{{\rm d}\zeta}{{\rm d}r} \tag{2.93}$$

Parameters such as $\frac{{\rm d}p}{{\rm d}\Psi}$ (pressure gradient), q (safety factor), $\hat{s}$ (magnetic shear) and ϵ (aspect ratio $= \frac{r}{R_{\rm mil}}$) are also used. Flux surfaces geometry are then given by :

$$R = R_{\rm mil} + r\cos\left(\theta + \arcsin\delta\sin\theta\right) \tag{2.94}$$
$$Z = Z_{\rm mil} + r\kappa\sin\left(\theta + \zeta\sin 2\theta\right) \tag{2.95}$$

and the radial derivatives of R and Z by :

$$\frac{\partial R}{\partial r} = \frac{{\rm d}R_{\rm mil}}{{\rm d}r} + \cos\left(\theta + \arcsin\delta\sin\theta\right) - s_\delta\cos\theta\sin\left(\theta + \arcsin\delta\sin\theta\right) \tag{2.96}$$
$$\frac{\partial Z}{\partial r} = \frac{{\rm d}Z_{\rm mil}}{{\rm d}r} + \kappa\sin\left(\theta + \zeta\sin 2\theta\right) + s_\kappa\sin\left(\theta + \zeta\sin 2\theta\right) + 2\kappa s_\zeta\cos\left(2\theta\right)\cos\left(\theta + \zeta\sin 2\theta\right) \tag{2.97}$$

The pressure gradient can be computed from different parameters through the input *gradp_type*. The input list is also defined in the file input.dat.sample.

The normalisation is made such that:

$$R_N = 1 + \epsilon\cos(\theta + \arcsin\delta\sin\theta) \tag{2.98}$$
$$F_N = 1 \tag{2.99}$$

We have then $R_{\rm ref} = R_{\rm mil}$ and $B_{\rm ref} = B_t(R_{\rm mil})$.
The normalisation for the plasma pressure gradient is given here using :

$$\left(\frac{{\rm d}p}{{\rm d}\Psi}\right)_N = \frac{2\mu_0 R_{\rm ref}^2}{B_{\rm ref}}\frac{{\rm d}p}{{\rm d}\Psi} \tag{2.100}$$

The normalised pressure gradient used in the code can be given by :

$$\left(\frac{{\rm d}p}{{\rm d}\Psi}\right)_N = -\alpha\frac{(2\pi^2)}{(\partial_\Psi V)_N}\left(\frac{V_N}{2\pi^2}\right)^{-1/2} \tag{2.101}$$
$$\left(\frac{{\rm d}p}{{\rm d}\Psi}\right)_N = \frac{\alpha_{MHD}}{\epsilon^2}\left(\frac{{\rm d}\Psi}{{\rm d}\epsilon}\right)_N \tag{2.102}$$
$$\left(\frac{{\rm d}p}{{\rm d}\Psi}\right)_N = \frac{\beta'}{2\frac{{\rm d}\Psi_N}{{\rm d}\epsilon}} \tag{2.103}$$

with α taken from equation 42 in [37], α_{MHD} from equation 141 [38] and β' is defined in section 9.3.6.
For centrifugal effects on the pressure gradient go to equation 2.140.
The equations used below are not normalised however they are in the code.
In the coordinate system (r, θ, φ) where the magnetic flux surfaces are described, the elements of the metric

tensor are :

$$\begin{aligned}
g_{rr} &= \left(\frac{\partial R}{\partial r}\right)^2 + \left(\frac{\partial Z}{\partial r}\right)^2, && (2.104)\\
g_{r\theta} &= \frac{\partial R}{\partial r}\frac{\partial R}{\partial \theta} + \frac{\partial Z}{\partial r}\frac{\partial Z}{\partial \theta} && (2.105)\\
g_{\theta\theta} &= \left(\frac{\partial R}{\partial \theta}\right)^2 + \left(\frac{\partial Z}{\partial \theta}\right)^2, && (2.106)\\
g_{\varphi\varphi} &= R^2 && (2.107)
\end{aligned}$$

The contravariant metric tensor, used to calculate the contravariant metric tensor of the GKW basis, is deduced from the relation $g_{ij} \cdot g^{ij} = I$ with I the identity matrix.

Mercier-Luc coordinate system

An expansion of the poloidal flux Ψ is needed. A coordinate system (introduced by Mercier-Luc) is defined such that the expansion is easy to use (orthogonal system (ρ, l, φ)). In addition to the plasma shape, the

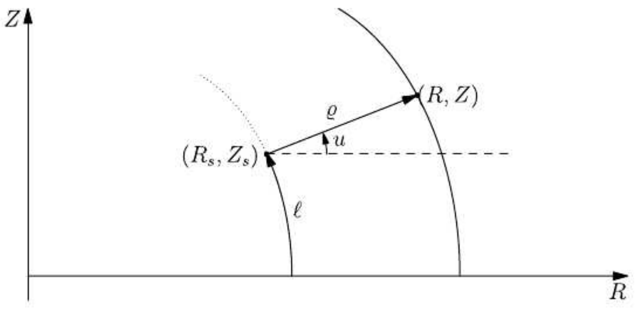


Figure 2.2: Mercier-Luc coordinate system....from [38]

gradient of the plasma pressure must also be specified, which in GKW can be done independently for the geometry, or by coupling to β'.

$$\begin{aligned}
\Psi &= \Psi_s + \rho\Psi_1 + \rho^2\Psi_2 && (2.108)\\
R &= R_s + \rho\cos u && (2.109)\\
Z &= Z_s + \rho\sin u && (2.110)
\end{aligned}$$

with

$$\begin{aligned}
\cos u &= \frac{\partial Z_s}{\partial l} && (2.111)\\
\sin u &= -\frac{\partial R_s}{\partial l} && (2.112)
\end{aligned}$$

where the subscript s means the value on the surface of consideration.
The metric elements are:

$$g_{\rho\rho} = 1 \tag{2.113}$$

$$g_{ll} = \left(1 + \frac{\rho}{r_c}\right) \tag{2.114}$$

$$g_{\varphi\varphi} = R^2 \tag{2.115}$$

The Jacobian is then:

$$J_\rho = R\left(1 + \frac{\rho}{r_c}\right) \tag{2.116}$$

with r_c being the curvature radius :

$$r_c = g_{\theta\theta}^{3/2}\left(\frac{\partial R}{\partial\theta}\frac{\partial^2 Z}{\partial\theta^2} - \frac{\partial Z}{\partial\theta}\frac{\partial^2 R}{\partial\theta^2}\right)^{-1} \tag{2.117}$$

To switch from (ρ, l, φ) to (r, θ, φ) the following relations are important (all quantities are taken on the flux surface) :

$$\frac{\partial\rho}{\partial r} = \cos u\frac{\partial R}{\partial r} + \sin u\frac{\partial Z}{\partial r} \tag{2.118}$$

$$\frac{\partial l}{\partial r} = \cos u\frac{\partial Z}{\partial r} - \sin u\frac{\partial R}{\partial r} \tag{2.119}$$

$$\frac{\partial\rho}{\partial\theta} = 0 \tag{2.120}$$

$$\frac{\partial l}{\partial\theta} = \sqrt{g_{\theta\theta}} \tag{2.121}$$

From the definition of the poloidal magnetic field:

$$B_p^2 = |\nabla\varphi|^2|\nabla\Psi|^2 = \frac{\Psi_1^2}{R_s^2} \tag{2.122}$$

one finds

$$\Psi_1 = R_s B_{ps} \tag{2.123}$$

Ψ_2 is obtained from the Grad-Shafranov equation for the expansion of Ψ:

$$R\nabla\cdot\frac{\nabla\Psi}{R^2} = -\mu_0 R^2 p'(\Psi) - FF'(\Psi) \tag{2.124}$$

with ' corresponding to the derivative toward the poloidal magnetic flux. One can easily obtain:

$$\Psi_2 = \frac{1}{2}\left(B_p\cos u - \frac{B_p R}{r_c} - \mu_0 R^2 p' - FF'\right) \tag{2.125}$$

From this point we see that we have other parameters : F (see the normalisation section), F', p', $\frac{d\Psi}{dr}$ that we need to link to our first set of parameters. The relation for F' will be expressed later (it is not trivial).

Relations for p' and $\frac{d\Psi}{dr}$

$\frac{d\Psi}{dr}$ is obtained from the definition of q:

$$q = \frac{1}{2\pi}\int_0^{2\pi}\frac{\mathbf{B}\cdot\nabla\varphi}{\mathbf{B}\cdot\nabla\theta}\mathrm{d}\theta \tag{2.126}$$

with

$$\mathbf{B} = s_B F \nabla\varphi + s_J \nabla\varphi \times \nabla\Psi \tag{2.127}$$

It follows :

$$q = s_B s_J \frac{F}{2\pi \frac{\mathrm{d}\Psi}{\mathrm{d}r}} \int_0^{2\pi} \frac{J_r}{R^2} \mathrm{d}\theta \tag{2.128}$$

yields

$$\frac{\mathrm{d}\Psi}{\mathrm{d}r} = s_B s_J \frac{F}{2\pi q} \int_0^{2\pi} \frac{J_r}{R^2} \mathrm{d}\theta \tag{2.129}$$

Numerical integrals are implemented using Simpson's method.
p' is given from the relation in [37]:

$$\alpha = -\frac{2\frac{\partial V}{\partial \Psi}}{(2\pi)^2} \left(\frac{V}{2\pi^2 R_{\mathrm{mil}}}\right)^{1/2} \mu_0 p' \tag{2.130}$$

V is the volume defined by the magnetic flux surface :

$$V = 2\pi S R_G \tag{2.131}$$

with

$$R_G = \frac{\int_0^{2\pi} R\sqrt{(\frac{\partial R}{\partial \theta})^2 + (\frac{\partial Z}{\partial \theta})^2}\mathrm{d}\theta}{\int_0^{2\pi} \sqrt{(\frac{\partial R}{\partial \theta})^2 + (\frac{\partial Z}{\partial \theta})^2}\mathrm{d}\theta} \tag{2.132}$$

$$S = \int_0^{2\pi} Z \frac{\partial R}{\partial \theta} \tag{2.133}$$

$\frac{\partial V}{\partial \Psi}$ can easily be defined analytically by differentiating towards r and using $\frac{\mathrm{d}\Psi}{\mathrm{d}r}$

Effect of toroidal rotation on the magnetic equilibrium

Force balance equation for species a:

$$m_a n_a \left(\frac{\partial}{\partial t} + V_a \cdot \nabla\right) V_a = -\nabla p_a - \nabla \cdot \pi_a + e_a n_a (E + V_a \times B) + R_a \tag{2.134}$$

Viscosity $\nabla \cdot \pi_a$ and friction R_a are being neglected.
A two species model with singly (for now) charged ions is considered (trace species can be added but won't affect the magnetic equilibrium: $m_t n_t \ll m_i n_i$, i being the main ion species). Using $m_e \ll m_i$ and taking the dot product of force balance equation with $\mathbf{b}$, one obtains:

$$n_e = n_{R_0,e} \exp\left(\frac{e\Phi}{T_e}\right) \tag{2.135}$$

$$n_i = n_{R_0,i} \exp\left(\frac{-e\Phi}{T_i} + \frac{m_i \Omega^2 (R^2 - R_0^2)}{2T_i}\right) \tag{2.136}$$

T_i is supposed to be a flux function (might not be always the case [39])
Using Quasineutrality and $T_e \sim T_i$:

$$e\phi = \frac{T_e}{T_e + T_i}\frac{m_i}{T_i}\frac{1}{2}\Omega^2\left(R^2 - R_0^2\right) \tag{2.137}$$

The total pressure is then:

$$p = Tn\exp\left(\frac{m_i\Omega^2}{2T}(R^2 - R_0^2)\right) \tag{2.138}$$

with $T = T_e + T_i$.
Furthermore the Grad-Shafranov is modified (Ψ and R^2 are regarded as independant variables [39]):

$$R^2\nabla\cdot\left(\frac{\nabla\Psi}{R^2}\right) = -\mu_0 R^2\frac{\partial p\left(\Psi, R^2\right)}{\partial\Psi} - FF' \tag{2.139}$$

In normalised units one gets

$$\begin{aligned}\left(\frac{\partial p}{\partial\Psi}\right)_N = \frac{1}{2}\Bigg[\beta' + \beta_N n_R m_{R,i}\Bigg(\left(R_N^2 - R_0^2\right)\Bigg(-2u_N'\Omega + \frac{T_{R,e}\frac{R}{L_{T_e}} + T_{R,i}\frac{R}{L_{T_i}}}{(T_{R,e} + T_{R,i})}\Bigg)\\ -R_0\frac{\partial R_0}{\partial\epsilon}\Omega^2\Bigg)\Bigg)\Bigg]\left(\frac{d\psi}{d\Psi}\right)_N\exp\left(\frac{m_{R,i}\Omega_N^2}{(T_{R,e} + T_{R,i})}\left(R_N^2 - R_0^2\right)\right)\end{aligned} \tag{2.140}$$

From this definition, it enters normalised Grad-Shafranov equation through:

$$\left(R^2\nabla\cdot\left(\frac{\nabla\Psi}{R^2}\right)\right)_N = -R_N^2\left(\frac{\partial p\left(\Psi, R^2\right)}{\partial\Psi}\right)_N - F_N F_N' \tag{2.141}$$

β' is given in SPCGENERAL. β_N can be taken from SPCGENERAL but can also be given using the GEOM parameter *beta_rota_miller_type = 'geom'* and specifying the value in *beta_rota_miller*. The other parameters are taken from SPECIES and ROTATION namelists.

Modification of the curvature drift

The magnetic curvature can be recast as:

$$\kappa = -\mathbf{b}\times\left(\nabla\times\frac{\mathbf{B}}{B}\right) \tag{2.142}$$

$$\kappa = -\frac{1}{B}\mathbf{b}\times(\nabla\times\mathbf{B}) - \mathbf{b}\times\left[\nabla\left(\frac{1}{B}\right)\times\mathbf{B}\right] \tag{2.143}$$

$$\kappa = \frac{\mu_0 J\times B}{B^2} + \frac{\nabla_\perp\mathbf{B}}{B} \tag{2.144}$$

$$\tag{2.145}$$

We will now only consider:

$$\kappa_1 = \mu_0\frac{J\times B}{B^2} \tag{2.146}$$

since the term $\nabla_\perp B$ is not modified by rotation.

Without rotation

$$J\times B = \nabla p \tag{2.147}$$

which leads to the β' correction.

For strong toroidal rotation one obtains for a two fluid model with singly charged ions:

$$J \times B = -m_i n_i R\Omega^2 \nabla R + \frac{\partial p}{\partial \Psi}\frac{d\Psi}{d\psi}\nabla\psi + \frac{\partial p}{\partial R^2}\nabla R^2 \tag{2.148}$$

The curvature enters Vlasov equation through

$$(\kappa_1 \times \nabla f)\cdot \mathbf{b} = \frac{\mu_0}{B^2}\left[\left(m_i n_i \Omega^2 - 2\frac{\partial p}{\partial R^2}\right)\left(B s_B \frac{\partial f}{\partial x_\alpha}\mathcal{I}^\alpha\right) + 2\frac{\partial p}{\partial \Psi}\frac{d\Psi}{d\psi}\frac{\partial f}{\partial x_\beta}\mathcal{E}^{\psi\beta}\right] \tag{2.149}$$

Determination of ∇s

We start from the expression of the magnetic field to express the contravariant component B^s in the coordinate system (Ψ, l, φ).

$$\begin{aligned} B^s(\Psi) &= B\cdot\nabla s && (2.150)\\ &= B\cdot\nabla l\frac{\partial s}{\partial l} && (2.151)\\ \frac{\partial s}{\partial l} &= B^s\frac{1}{B\cdot\nabla l} && (2.152)\end{aligned}$$

Furthermore :

$$\int_0^L \frac{\partial s}{\partial l'}\mathrm{d}l' = 1 = B^s\int_0^L \frac{1}{B\cdot\nabla l'}\mathrm{d}l' \tag{2.153}$$

with L being the arclength corresponding to $\theta = 2\pi$. Hence

$$B^s = \frac{1}{\int_0^L \frac{1}{B\cdot\nabla l'}\mathrm{d}l'} \tag{2.154}$$

One gets the expression of s in this basis:

$$s = B^s\int_0^l \frac{1}{B\cdot\nabla l'}\mathrm{d}l' \tag{2.155}$$

Using the expansion of the magnetic flux:

$$\begin{aligned} B\cdot\nabla l &= s_B F\nabla\varphi\cdot\nabla l + s_j\nabla\varphi\times\left[(\Psi_1 + 2\rho\Psi_2)\nabla\rho + \rho\frac{\partial\Psi_1}{\partial l}\nabla l\right]\cdot\nabla l && (2.156)\\ B\cdot\nabla l &= s_j\frac{\Psi_1 + 2\rho\Psi_2}{J_\rho} && (2.157)\\ B\cdot\nabla l &= s_j\frac{\Psi_1 + 2\rho\Psi_2}{(R_s + \rho\cos u)(1 + \frac{\rho}{r_c})} && (2.158)\end{aligned}$$

Then

$$\frac{1}{B\cdot\nabla l} = \frac{s_j}{\Psi_1}\left[R_s + \rho\left(\cos u + \frac{R_s}{r_c} - 2R_s\frac{\Psi 2}{\Psi 1}\right)\right] \tag{2.159}$$

One can write ∇s as

$$\nabla s = \frac{\partial s}{\partial\Psi}\nabla\Psi + \frac{\partial s}{\partial l}\nabla l \tag{2.160}$$

We need then to determine $\frac{\partial s}{\partial \Psi}$. Using the Mercier-Luc expansion of Ψ one gets the second order equation in ρ:

$$\Psi_2\rho^2 + \Psi_1\rho + \Psi_s - \Psi = 0 \tag{2.161}$$

At $\Psi = \Psi_s$, $\rho = 0$. We have then ρ as a function of Ψ

$$\rho = \frac{-\Psi_1 + \sqrt{\Psi_1^2 - 4\Psi_2(\Psi_s - \Psi)}}{2\Psi_2} \tag{2.162}$$

For $\Psi = \Psi_s$, $\frac{\partial \rho}{\partial \Psi} = \frac{1}{\Psi_1}$. We define the two following integrals:

$$A_1(l) = \int_0^l s_j \frac{R_s}{\Psi_1} \mathrm{d}l' \tag{2.163}$$

$$A_2(l) = \int_0^l \frac{s_j}{\Psi_1}\frac{\partial \rho}{\partial \Psi}\left(\cos u + \frac{R_s}{r_c} - 2\frac{R_s\Psi_2}{\Psi_1}\right)\mathrm{d}l' \tag{2.164}$$

At $\Psi = \Psi_s$ we then have:

$$\frac{\partial s}{\partial \Psi} = \frac{A_2(l)}{A_1(L)} - \frac{A_2(L)A_1(l)}{A_1(L)^2} \tag{2.165}$$

∇s is then given on the flux surface by :

$$\nabla s = \left(\frac{\partial s}{\partial \Psi}\frac{\mathrm{d}\Psi}{\partial \mathrm{d}r} + \frac{\partial s}{\partial l}\frac{\partial l}{\partial r}\right)\nabla r + \frac{\partial s}{\partial l}\frac{\partial l}{\partial \theta}\nabla\theta \tag{2.166}$$

Determination of $\nabla\zeta$

The magnetic field can be written as:

$$\mathbf{B} = s_J 2\pi \nabla\Psi \times \nabla\zeta \tag{2.167}$$

Using the following expansion for the zeta coordinate [38]:

$$\zeta = \zeta_s(l) + \rho\zeta_1(l) - \frac{\varphi}{2\pi} + \mathcal{O}(\rho^2) \tag{2.168}$$

one gets

$$F = F_s + \rho\Psi_1 F_s' + \mathcal{O}(\rho^2) \tag{2.169}$$

$$\nabla\Psi = (\Psi_1 + 2\rho\Psi_2)\nabla\rho + \rho\frac{\partial \Psi_1}{\partial l}\nabla l + \mathcal{O}(\rho^2) \tag{2.170}$$

$$\nabla\zeta = \zeta_1\nabla\rho + \left(\frac{\partial \zeta_s}{\partial l} + \rho\frac{\partial \zeta_1}{\partial l}\right)\nabla l + \frac{\nabla\varphi}{2\pi} + \mathcal{O}(\rho^2) \tag{2.171}$$

From the two expressions of $\mathbf{B}$:

$$2s_J\pi\nabla\Psi \times \nabla\zeta = s_B F\nabla\varphi + s_J\nabla\Psi \times \nabla\varphi \tag{2.172}$$

dotting with $\nabla\varphi$ yields

$$\frac{2\pi s_J}{J_\rho}\left[(\Psi_1 + 2\rho\Psi_2)\left(\frac{\partial \zeta_s}{\partial l} + \rho\frac{\partial \zeta_1}{\partial l}\right) - \rho\zeta_1\frac{\partial \Psi_1}{\partial l}\right] = s_B\frac{F_s + \rho\Psi_1 F_s'}{(R_s + \rho\cos u)^2} \tag{2.173}$$

At zeroth order in ρ:

$$\frac{\partial \zeta_s}{\partial l} = s_J s_B \frac{F_s}{2R_s\Psi_1\pi} \tag{2.174}$$

at first order :

$$2s_J s_B \pi \left(\Psi_1 \frac{\partial \zeta_1}{\partial l} - \zeta_1 \frac{\partial \Psi_1}{\partial l} + 2\Psi_2 \frac{\partial \zeta_s}{\partial l} \right) = \frac{F_s}{R_s}\left(\frac{1}{r_c} - \frac{\cos u}{R_s}\right) + \frac{F_s' \Psi_1}{R_s} \tag{2.175}$$

hence

$$\Psi_1^2 s_J s_B \frac{\partial}{\partial l}\left(\frac{\zeta_1}{\Psi_1}\right) = \frac{1}{2\pi}\left[\frac{F_s}{R_s}\left(\frac{1}{r_c} - \frac{\cos u}{R_s}\right) + \frac{F_s'\Psi_1}{R_s}\right] - 2\Psi_2 \frac{\partial \zeta_s}{\partial l} \tag{2.176}$$

ζ_s and ζ_1 are then defined by:

$$\zeta_s(l) = s_J s_B \int_0^l \frac{F_s}{2\pi R_s \Psi_1} \mathrm{d}l' \tag{2.177}$$

$$\zeta_1(l) = s_J s_B \Psi_1 \int_0^l \left(\frac{1}{2\pi}\left[\frac{F_s}{R_s}\left(\frac{1}{r_c} - \frac{\cos u}{R_s}\right) + \frac{F_s'\Psi_1}{R_s}\right] - \frac{\Psi_2 F_s}{\pi R_s \Psi_1}\right) \frac{1}{\Psi_1^2} \mathrm{d}l' \tag{2.178}$$

replacing Ψ_2 by its value one can define the following integrals:

$$D_0(l) = \int_0^l \frac{F_s}{\pi \Psi_1^2 R_s}\left(\frac{1}{r_c} - \frac{\cos u}{R_s}\right) \mathrm{d}l' \tag{2.179}$$

$$D_1(l) = \int_0^l \left(\frac{F_s^2}{2\pi R_s \Psi_1^3} + \frac{1}{2\pi R_s \Psi_1}\right) \mathrm{d}l' \tag{2.180}$$

$$D_2(l) = \int_0^l \frac{\mu_0 R_s F_s}{2\pi \Psi_1^3} \mathrm{d}l' \tag{2.181}$$

$\zeta_1(l)$ can be then written in the form:

$$\zeta_1(l) = s_J s_B \Psi_1 \left(D_0(l) + D_1(l) F' + D_2(l) p'\right) \tag{2.182}$$

Warning: If the plasma rotation is taken into account through p', it will depend on R and therefore on l. p' is then in $D_2(l)$
F' has not yet been linked to the input parameters. Once we have F' we see that $\frac{\partial \zeta}{\partial l} = \zeta_s$ and $\frac{\partial \zeta}{\partial \rho} = \zeta_1$.
$\nabla \zeta$ is then given on the flux surface by:

$$\nabla \zeta = \left(\zeta_s \frac{\partial l}{\partial r} + \zeta_1 \frac{\partial \rho}{\partial r}\right) \nabla r + \zeta_s \frac{\partial l}{\partial \theta} \nabla \theta \tag{2.183}$$

Relation for F'

Using the expression of q and the specific boundary condition [38]:

$$\zeta(\Psi, 0, \varphi) = FIXME \tag{2.184}$$

one gets

$$\zeta(\Psi, 2\pi, \varphi) = -2\pi q(\Psi) \tag{2.185}$$

using the expansion of Ψ we have the following equation:

$$-2\pi(q_s + q_s' \rho \Psi_1) + \mathcal{O}(\rho^2) = \zeta_s(L) + \rho \zeta_1(L) + \mathcal{O}(\rho^2) \tag{2.186}$$

We then have:

$$F_s' = \frac{2\pi \hat{s} \frac{q}{r} - D_0(L) - D_2(L) p'}{D_1(L)} \tag{2.187}$$

Contravariant metric tensor of (r, ζ, s)

We can then deduce the contravariant metric tensor in the GKW basis.

$$g^{r\zeta} = g^{\zeta r} = \frac{\partial \zeta}{\partial r} g^{rr} + \frac{\partial \zeta}{\partial \theta} g^{r\theta} \tag{2.188}$$

$$g^{rs} = g^{sr} = \frac{\partial s}{\partial r} g^{rr} + \frac{\partial s}{\partial \theta} g^{r\theta} \tag{2.189}$$

$$g^{\zeta\zeta} = \left(\frac{\partial \zeta}{\partial r}\right)^2 g^{rr} + \left(\frac{\partial \zeta}{\partial \theta}\right)^2 g^{\theta\theta} + 2\frac{\partial \zeta}{\partial r}\frac{\partial \zeta}{\partial \theta} g^{r\theta} + \frac{1}{4\pi^2} g^{\varphi\varphi} \tag{2.190}$$

$$g^{ss} = \left(\frac{\partial s}{\partial r}\right)^2 g^{rr} + \left(\frac{\partial s}{\partial \theta}\right)^2 g^{\theta\theta} + 2\frac{\partial s}{\partial r}\frac{\partial s}{\partial \theta} g^{r\theta} \tag{2.191}$$

$$g^{\zeta s} = g^{s\zeta} = \left(\frac{\partial \zeta}{\partial r}\frac{\partial s}{\partial r}\right) g^{rr} + \left(\frac{\partial \zeta}{\partial \theta}\frac{\partial s}{\partial \theta}\right) g^{\theta\theta} + \left(\frac{\partial \zeta}{\partial r}\frac{\partial s}{\partial \theta} + \frac{\partial \zeta}{\partial \theta}\frac{\partial s}{\partial r}\right) g^{r\theta} \tag{2.192}$$

Derivatives of the magnetic field strength

To obtain the derivatives of the magnetic field strength B we start from [38]:

$$B_p^2 = |\nabla\varphi|^2 |\nabla\Psi|^2 \tag{2.193}$$

$$B_t^2 = \left|\frac{F(\Psi)}{R}\right| \tag{2.194}$$

Using the expansion of the magnetic flux one obtains:

$$R^2 B_p^2 = (\Psi_1 + 2\rho\Psi_2)^2 g^{\rho\rho} + \left(\rho \frac{\partial \Psi_1}{\partial l}^2 g^{ll}\right) \tag{2.195}$$

$$= \Psi_1^2 + 4\rho\Psi_1\Psi_2 + \mathcal{O}(\rho^2) \tag{2.196}$$

$$R^2 B_t^2 = F_s^2 + 2\rho\Psi_1 F_s F_s' + \mathcal{O}(\rho^2) \tag{2.197}$$

Using now the expansion on R and differentiating by ρ and l

$$\frac{\partial B_p^2}{\partial \rho} = \frac{1}{R_s^2}\left(4\Psi_1\Psi_2 - 2\frac{\Psi_1^2}{R_s}\cos u\right) \tag{2.198}$$

$$\frac{\partial B_t^2}{\partial \rho} = \frac{1}{R_s^2}\left(2F_s F_s'\Psi_1 - 2\frac{F_s^2}{R_s}\cos u\right) \tag{2.199}$$

$$\frac{\partial B_p^2}{\partial l} = \frac{1}{R_s^2}\left(2\Psi_1\frac{\partial \Psi_1}{\partial l} - 2\frac{\partial R_s}{\partial l}\frac{\Psi_1^2}{R_s}\right) \tag{2.200}$$

$$\frac{\partial B_t^2}{\partial l} = -\frac{2}{R_s^2}\frac{\partial R_s}{\partial l}\frac{F^2}{R_s} \tag{2.201}$$

We can then deduce the derivatives of B by summing the previous equations

$$\frac{\partial (B^2)}{\partial \rho} = \frac{1}{R_s^2}\left(4\Psi_1\Psi_2 - 2\cos u\frac{\Psi_1}{R_s^2} + 2F_s F_s'\Psi_1 - 2\frac{F_s^2}{R_s\cos u}\right) \tag{2.202}$$

$$\frac{\partial (B^2)}{\partial l} = \frac{1}{R_s^2}\left(2\Psi_1\frac{\partial \Psi_1}{\partial l} - 2\frac{\partial R_s}{\partial l}\left(\frac{\Psi_1^2}{R_s} - \frac{F^2}{R_s}\right)\right) \tag{2.203}$$

A successful benchmark of the Miller geometry against GS2 is presented in Sec. 11.1.

2.2.4 Sheared slab geometry

For consistency with the way the geometry and boundary conditions are treated in the code here we derive Hamada field aligned slab coordinates.

We require that $\boldsymbol{B} \cdot \nabla s = B^s$ is a flux function.

The normalised coordinate along the field line is defined similary to the toroidal case, where:

$$s = \frac{z}{l_s} \tag{2.204}$$

Where s varied between -1/2 and 1/2. The components of the magnetic field are written as:

$$\begin{aligned} \boldsymbol{B} \cdot \nabla s &= \boldsymbol{B} \cdot \nabla x \frac{\partial s}{\partial x} + \boldsymbol{B} \cdot \nabla y \frac{\partial s}{\partial y} + \boldsymbol{B} \cdot \nabla z \frac{\partial s}{\partial z} && (2.205) \\ \boldsymbol{B} \cdot \nabla z &= B^z \frac{\partial s}{\partial z} = B^z / l_z = B^s && (2.206) \\ B^s &= B^z / l_z && (2.207) \end{aligned}$$

Where l_x,l_y and l_z are the computational box sizes in each of the three directions.

The perpendicular component has the form:

$$\boldsymbol{B} \cdot \nabla \zeta = \boldsymbol{B} \cdot \nabla y \frac{\partial \zeta}{\partial y} + \boldsymbol{B} \cdot \nabla z \frac{\partial \zeta}{\partial z} = 0 \tag{2.208}$$

Which gives:

$$B^y \frac{\partial \zeta}{\partial y} + B^z \frac{\partial \zeta}{\partial z} = 0 \tag{2.209}$$

If we say that the binormal coordinate is a linear function of the x and z directions

$$\zeta = \alpha y + \beta z \tag{2.210}$$

we choose $\alpha = -1$, which gives:

$$\beta = \frac{B^y}{B^z} \tag{2.211}$$

The field aligned, sheared slab coordinate transform from cartesian coordinates x,y,z can be written as:

$$\begin{aligned} \psi &= x \\ \zeta &= -y + \frac{B^y}{B^z} z \\ s &= z && (2.212) \end{aligned}$$

The negative sign in the definition of ζ is used to obtain the same right handed set (ψ, ζ, s) as in the toroidal geometry. CAUTION: This means that the set (x, y, z) is left handed. Here y corresponds to γ.

We expand the y component ($B_0^y = 0$) giving:

$$B^y = (B_y^{'})\psi \tag{2.213}$$

where the prime denotes the radial derivative, and as such we define

$$\hat{s} = \frac{1}{\boldsymbol{B} \cdot \nabla z} \frac{\partial}{\partial x} (\boldsymbol{B} \cdot \nabla y) \tag{2.214}$$

When we normalise the y direction, we get a perpendicular component of the form:

$$\zeta_N = -\frac{y}{l_y} - \hat{s}_N \frac{l_x l_z}{l_y} \psi s \tag{2.215}$$

(signs ???) where

$$\hat{s}_N = \frac{l_x l_z}{l_y} \hat{s} \tag{2.216}$$

This is the same as having the magnetic field of the form:

$$\boldsymbol{B} = B_0(\hat{z} + \hat{s}x\hat{y}) \tag{2.217}$$

With this form it can be shown that:

$$\boldsymbol{b} \cdot \nabla = \frac{\partial}{\partial s} \tag{2.218}$$

The differentials have the form:

$$\begin{aligned}
\nabla\psi &= \frac{\partial x'}{\partial x}\hat{x} = \hat{x} && (2.219)\\
\nabla\zeta &= -\hat{y} + \hat{s}z\hat{x} + \hat{s}x\hat{z} && (2.220)\\
\nabla s &= \hat{z} && (2.221)
\end{aligned}$$

In the slab model, the input parameters ϵ and q do not affect the geometry but must both be set to 1 for correct k_x mode spacing with *mode_box=T*. It can be seen that to obtain correspondance with the toroidal geometries, we must set $q = \hat{s}\psi$.

Parallel Boundary Conditions

The slab is always defined to be periodic in the 'radial' (x) and 'poloidal' (y) directions. We may also wish to maintain the parallel (z) periodicity and shear boundary conditions as in the toroidal case (Sec. 2.3), i.e.

$$f(\psi, \zeta, s + 1/2) = f(\psi, \zeta, s - 1/2) \tag{2.222}$$

The mode amplitudes

$$A \exp \imath k_\zeta \zeta = A_{-1/2} \exp \imath k_\zeta \zeta_0 - \imath k_\zeta 1/2 \hat{s}x = A_{1/2} \exp \imath k_\zeta \zeta_0 + \imath k_\zeta 1/2 \hat{s}x \tag{2.223}$$

and as such

$$\Delta k_x = k_\zeta \frac{\hat{s}}{i_k} \tag{2.224}$$

where i_k is an integer. To allow the slab periodicity condition to be equivalent with the condition in toroidal geometry , we must set $q = 1$ and $\epsilon = 1$ in the slab case (c.f eq. 2.255). This makes the toroidal $\hat{s}$ equivalent to the one defined above, allowing the same numerical implementation for all geometries. The periodic slab is selected in GKW as the geometry 'slab_periodic', for use with mode box only. See also Hammett et Al 1993.

Alternatively the slab model can be chosen to be not periodic but infinite in the parallel direction. In GKW the 'slab' model has no periodicity (the boundary condition is flappy, the same as at the "end" of the toroidal field line). For 'slab' only, nperiod > 1 can therefore be run with mode box. Without mode box, 'slab' and 'slab_periodic' are identical.

Geometry tensors

WARNING: The signs in this section are incorrect and need to be fixed. The S_B and S_j were added by anaolgy with the toroidal case and are not kept in the derivation above.

We now calculate the geometry tensors, first the diagonal terms are as follows,

$$\begin{aligned} g^{\psi\psi} &= \nabla\psi\cdot\nabla\psi = 1 && (2.225)\\ g^{\zeta\zeta} &= \nabla\zeta\cdot\nabla\zeta = 1 + (\hat{s}s)^2 + (sx)^2 = 1 + (\hat{s}s)^2 && (2.226)\\ g^{ss} &= \nabla s\cdot\nabla s = 1 && (2.227)\\ & && (2.228) \end{aligned}$$

then the off diagonal:

$$\begin{aligned} g^{\psi\zeta} &= \nabla\psi.\nabla\zeta = -s_B s_j \hat{s}s = g^{\zeta\psi} && (2.229)\\ g^{\psi s} &= g^{s\psi} = 0 && (2.230)\\ g^{\zeta s} &= g^{s\zeta} = -\hat{s}x = 0 && (2.231) \end{aligned}$$

finally the derived tensors

$$\begin{aligned} & \mathcal{D} = \mathcal{G} = \mathcal{H} = \mathcal{I} = 0 && (2.232)\\ \mathcal{F} = s_j, \qquad & \mathcal{E}^{\psi\zeta} = -\mathcal{E}^{\zeta\psi} = -\frac{s_j}{2} && (2.233)\\ & && (2.234) \end{aligned}$$

with all other components of $\mathcal{E} = 0$.

We maintain the local approximation and the above tensors are evaluated at the centre of the computational domain, which is set to $x = 0$.

2.2.5 General geometry

The metric elements for general toroidal MHD equilibria can be obtained from the CHEASE code [42]. The sign of the magnetic field and plasma current are not taken into account into CHEASE which always consider that $\mathbf{B}$ and $\mathbf{j}$ are in the direction opposite to $\nabla\varphi$ (here $\nabla\varphi = \nabla\varphi_{\mathrm{GKW}}$, as everywhere in this document). The sign of $\mathbf{B}$ and $\mathbf{j}$ are therefore specified in GKW, in the *GEOM* namelist using the variables *SIGNB* and *SIGNJ* defined as in the previous section by $s_{\mathrm{B}} = \mathrm{sign}(\mathbf{B}\cdot\nabla\varphi)$ and $s_{\mathrm{j}} = \mathrm{sign}(\mathbf{j}\cdot\nabla\varphi)$. The flux surface of interest is also specified in the *GEOM* namelist using the *EPS* parameter. Depending on the value (1 or 2) of the *EPS_TYPE* parameter, *EPS* is either the local inverse aspect ratio (i.e. the ψ coordinate of GKW) or ρ_Ψ the square root of the normalised poloidal flux ($\rho_\Psi = 0$ on the axis and $\rho_\Psi = 1$ on the last closed flux surface). When CHEASE is used, the absolute value of the safety factor and magnetic shear (used for the periodicity condition and to set the radial wave vector grid) is taken from the equilibrium reconstruction and the values specified in the namelist *GEOM* are not read.

When wished, the values of β and β' consistent with the MHD equilibrium can be used as inputs in GKW as detailed in Sec. 9.3.6.

For the CHEASE geometry, $R_{\mathrm{ref}} = R0EXP$ which is specified by CHEASE. It may be near the magnetic axis, but you should always check the value used in the *hamada.dat* output of CHEASE. The seperate document RunChease.tex contains more information on running CHEASE for GKW.

The $R_0 = R_{\mathrm{axis}}$ option for the centrifugal terms is not quite exact for the general geometry, since an s point on the axis may not exist. The nearest s point to $\theta = \pi/2$ is taken (the TOP intersection of the flux surface with the magnetic axis). For heavy impurities, this approximation could be innaccurate. The $R_0 = R_{\rm LFS}$ option is exact for $R_0 = R(s = 0)$, which may not be at $\theta = 0$ The value of the poloidal angle used is output to screen, and the actual R_0 used is written to `geom.dat`.

2.2.6 Summary of the sign conventions in GKW

In GKW, the cylindrical coordinate system (R, Z, φ) is right handed, which means that φ is increasing clockwise when the torus is viewed from above.
The toroidal rotation is defined positive for a plasma flowing in the direction of $\mathbf{B}$.
The mode frequency is defined positive for a perturbation evolving in the direction opposite to $\nabla\zeta$. This corresponds to the ion ∇B drift direction if $s_{\rm j} = 1$ and to the electron ∇B drift direction if $s_{\rm j} = -1$. See also Fig. 10.1
Coordinate system:

- $\psi = \varepsilon = (R_{\max} - R_{\min})/(2R_{\rm ref})$ is always increasing from the plasma center to the plasma edge.
- s is always increasing upwards from the low field side midplane. It is zero at the height of the magnetic axis, on the low field side midplane.
- ζ is always increasing in the direction opposite to φ (i.e. anticlockwise when viewed from above) at constant ψ and s. The direction of $\nabla\zeta$ in the poloidal plane is given by $\text{sign}(\nabla s \cdot \nabla\zeta) = \text{sign}(\nabla\zeta \cdot \nabla\theta) = s_{\rm B} s_{\rm j}$

$$\begin{aligned}
\text{sign}(\mathbf{B} \cdot \nabla\varphi) &\equiv s_{\rm B} && (2.235)\\
\text{sign}(\mathbf{j} \cdot \nabla\varphi) &\equiv s_{\rm j} && (2.236)\\
\text{sign}(\mathbf{B} \cdot \nabla\theta) &= s_{\rm j} && (2.237)\\
\text{sign}(\mathbf{B} \cdot \nabla s) &= s_{\rm j} && (2.238)\\
\text{sign}(\nabla\varphi \cdot \nabla\zeta) &= -1 && (2.239)\\
\text{sign}(\nabla s \cdot \nabla\theta) &= 1 && (2.240)\\
\text{sign}(\nabla\theta \cdot \nabla\zeta) &= s_{\rm B} s_{\rm j} && (2.241)\\
\text{sign}(B_\theta \nabla\theta \cdot \nabla\zeta) &= s_{\rm B} && (2.242)\\
\text{sign}(\nabla s \cdot \nabla\zeta) &= s_{\rm B} s_{\rm j} && (2.243)\\
\mathbf{B} \cdot \nabla\zeta &= 0 && (2.244)\\
\boldsymbol{\Omega} &= -s_{\rm B}\Omega\nabla z && (2.245)\\
u &= R\Omega && (2.246)
\end{aligned}$$

2.3 Spectral representation

GKW uses a combination of finite difference techniques and pseudo-spectral methods (similar to GS2 [11, 12]). Due to the homogeneous nature of the turbulence all perturbed quantities in the plane perpendicular to the magnetic field are represented by a sum over Fourier modes

$$f(\psi, \zeta, s) = \sum_{k_\zeta k_\psi} \hat{f}(k_\psi, k_\zeta, s) \exp[{\rm i}k_\zeta \zeta/\rho_* + {\rm i}k_\psi \psi/\rho_*] = \mathcal{T}^{-1}(\hat{f}), \qquad (2.247)$$

where the normalised Larmor radius (ρ_*) has been added for convenient normalisation so that

$$\rho_* \frac{\partial}{\partial x_\alpha} \rightarrow {\rm i}k_\alpha. \qquad (2.248)$$

Because the distribution function is real, the Fourier amplitude of the mode with wave vector (k_ζ, k_ψ) is equal to the complex conjugate of the mode with the wave vector $(-k_\zeta, -k_\psi)$. Internally, the code, therefore, calculates the evolution of the modes with $k_\zeta \geq 0$ only, whereas both positive as well as negative radial wave vectors (k_ψ) are used. For $k_\zeta > 0$ both signs of the wave vector are, therefore, not represented internally, whereas for $k_\zeta = 0$ they are. Consequently the real valued distribution function, when expressed in the modes evaluated in the code is

$$\begin{aligned} f(\psi, \zeta, s) &= \sum_{k_\zeta > 0, k_\psi} \left[\hat{f}(k_\psi, k_\zeta, s) \exp[\mathrm{i}k_\zeta \zeta/\rho_* + \mathrm{i}k_\psi \psi/\rho_*] + \hat{f}^\dagger(k_\psi, k_\zeta, s) \exp[-\mathrm{i}k_\zeta \zeta/\rho_* + \mathrm{i}k_\psi \psi/\rho_*] \right] \\ &\quad + \sum_{k_\zeta = 0, k_\psi} \hat{f}(k_\psi, k_\zeta = 0, s) \exp[\mathrm{i}k_\psi \psi/\rho_*], \end{aligned} \tag{2.249}$$

where the dagger denotes the complex conjugate. For the correct calculation of quadratic quantities like the fluxes, for instance, this representation must be considered and can lead to additional factors of two, when compared with the straightforward Fourier representation (see also section 10.4).

In what follows a $\hat{\cdot}$ indicates the Fourier representation of a quantity and $\mathcal{T}$ and $\mathcal{T}^{-1}$ represent the forward and inverse Fourier transform operations, respectively. The use of Fourier harmonics means that the boundary conditions in the perpendicular plane are periodic. The periodicity of the torus is, however, not automatically satisfied. The condition of toroidal periodicity can be formulated as

$$f(\psi, \zeta + 1, s) = f(\psi, \zeta, s) \quad \rightarrow \quad \frac{k_\zeta}{2\pi\rho_*} = N, \tag{2.250}$$

where N is an integer. Since ρ_* is small, this condition can in general be satisfied with very small changes to k_ζ or ρ_*. In the local limit employed here, the final equations are independent of ρ_* and it is assumed that the relation above is satisfied. The poloidal periodicity translates to

$$f(\psi, \zeta + q/2, 1/2) = f(\psi, \zeta - q/2, -1/2). \tag{2.251}$$

This condition translates to

$$\begin{aligned} &\sum_{\mathbf{k}} \hat{f}(k_\psi, k_\zeta, \frac{1}{2}) \exp\left[\frac{\mathrm{i}k_\zeta}{\rho_*} + \frac{\mathrm{i}k_\psi \psi}{\rho_*} + \frac{\mathrm{i}qk_\zeta}{2\rho_*}\right] \\ = &\sum_{\mathbf{k}} \hat{f}(k_\psi, k_\zeta, -\frac{1}{2}) \exp\left[\frac{\mathrm{i}k_\zeta}{\rho_*} + \frac{\mathrm{i}k_\psi \psi}{\rho_*} - \frac{\mathrm{i}qk_\zeta}{2\rho_*}\right] \end{aligned} \tag{2.252}$$

Expanding the safety factor around a reference value q_R (value at the centre of the radial domain)

$$\frac{qk_\zeta}{\rho_*} = \frac{q_R k_\zeta}{\rho_*} + k_\zeta \frac{\partial q}{\partial \psi} \frac{\psi}{\rho_*} + \frac{1}{2} k_\zeta \rho_* \frac{\partial^2 q}{\partial \psi^2} \left(\frac{\psi}{\rho_*}\right)^2, \tag{2.253}$$

and neglecting the second derivative correction (which gives only a ρ_* variation over the size of the box) one can see that this condition can be satisfied if also $q_R k_\zeta / 2\pi\rho_*$ is assumed to be an integer and if

$$\hat{f}(k_\psi, k_\zeta, \frac{1}{2}) = \hat{f}(k_\psi + k_\zeta \frac{\partial q}{\partial \psi}, k_\zeta, -\frac{1}{2}), \tag{2.254}$$

i.e. on the boundary of the s domain one simply connects the mode with the appropriate higher k_ψ mode. In the code, this boundary condition is implemented in the routine *connect_parallel*. This formulation is close to the 'ballooning transform' [43], and has previously been employed in GS2 [11, 12]. The above boundary condition for the Fourier amplitudes implies that a convenient choice for the spacing of the k_ψ modes in any discrete Fourier representation is

$$\Delta k_\psi = k_{\zeta,\min} \frac{\partial q}{\partial \psi} \frac{1}{i_k}, \tag{2.255}$$

where i_k is some integer (*IKXSPACE* in the code). The integer i_k allows for a control over the spacing between the radial modes which otherwise would be dictated by the magnetic shear.

In order to specify the wave vector in normalised form, the expression

$$(k_\theta \rho_{\rm ref})^2 = g^{\zeta\zeta} k_\zeta^2 \tag{2.256}$$

is evaluated at the low field side midplane ($s = 0$) to determine k_ζ from the value of $k_\theta \rho_{\rm ref}$ given as an input. Note that $\rho_{\rm ref}$ is defined on the flux surface at the major radius of the magnetic axis. Here k_θ is an unnormalised dimensional quantity, but k_ζ and $g^{\zeta\zeta}$ are normalised.

The k_θ notation used is historical but sometimes misleading; strictly speaking it would be more accurate to describe k_θ as "$k_\perp$ at the low field side for a radial mode". It is related to the toroidal mode number n by

$$n = \frac{k_\theta \rho_{\rm ref}}{2\pi \rho_* \sqrt{g^{\zeta\zeta}|_{s=0}}} = \frac{k_\zeta}{2\pi\rho_*} \tag{2.257}$$

where $\sqrt{g^{\zeta\zeta}|_{s=0}}$ is output in `geom.dat` as *kthnorm*. Note that only in the $s - \alpha$ geometry does $k_\theta \rho_* = nq/r$, in other cases particular care should be taken when comparing with other codes, with the toroidal mode number the least ambiguous quantity for comparison.

The full perpendicular wave vector is given by

$$k_\perp^2 = g^{\alpha\beta} k_\alpha k_\beta. \tag{2.258}$$

One of the main advantages of the Fourier representation is that the gyro-average becomes an algebraic operation

$$\widehat{\langle \phi \rangle} = J_0(k_\perp \rho) \widehat{\phi}, \tag{2.259}$$

where J_0 is the Bessel function and

$$(k_\perp \rho)^2 = \frac{m_R T_R}{Z^2 B^2} g^{\alpha\beta} k_\alpha k_\beta. \tag{2.260}$$

In the equation above Z is the charge number of the species considered. GKW can run with an arbitrary number of species, which have arbitrary mass and charge.

2.4 The complete set of equations

In this section we give the complete set of equations for a collisionless rotating plasma based on the material introduced in the previous sections. The gyrokinetic equation can be written in the form

$$\frac{\partial g}{\partial t} = \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV} + \mathrm{V} + \mathrm{VI} + \mathrm{VII} + \mathrm{VIII}, \tag{2.261}$$

with

$$\begin{aligned}
\mathrm{I} &= -v_\parallel \mathbf{b}\cdot\nabla f \to -v_R v_\parallel \mathcal{F}\frac{\partial \hat{f}}{\partial s},\\
\mathrm{II} &= -\mathbf{v}_D\cdot\nabla f \to\\
&\quad -\frac{\mathrm{i}}{Z}\left[T_R E_D \mathcal{D}^\alpha + T_R v_\parallel^2 \beta' \mathcal{E}^{\psi\alpha} + 2 m_R v_R v_\parallel \Omega \mathcal{H}^\alpha + m_R \Omega^2 I^\alpha + Z\mathcal{E}^{\beta\alpha}\frac{\partial \Phi}{\partial x_\beta}\right] k_\alpha \hat{f} + \\
&\quad -\frac{\rho_*}{Z}\left[T_R E_D \mathcal{D}^s + T_R v_\parallel^2 \beta' \mathcal{E}^{\psi s} + 2 m_R v_R v_\parallel \Omega \mathcal{H}^s + m_R \Omega^2 I^s + Z\mathcal{E}^{\beta s}\frac{\partial \Phi}{\partial x_\beta}\right]\frac{\partial \hat{f}}{\partial s}
\end{aligned} \tag{2.262}$$

$$\begin{aligned}
\mathrm{III} &= -\mathbf{v}_\chi\cdot\nabla g \rightharpoonup -\rho_*^2 \frac{\partial \chi}{\partial x_\beta}\mathcal{E}^{\beta\alpha}\frac{\partial g}{\partial x_\alpha} = \rho_*^2 \mathcal{E}^{\psi\zeta}\left(\frac{\partial \chi}{\partial \zeta}\frac{\partial g}{\partial \psi} - \frac{\partial g}{\partial \zeta}\frac{\partial \chi}{\partial \psi}\right)\\
&\quad \rightharpoonup \mathcal{T}\Big(\mathcal{E}^{\psi\zeta}\left[\mathcal{T}^{-1}(ik_\zeta\hat{\chi})\mathcal{T}^{-1}(ik_\psi\hat{g}) - \mathcal{T}^{-1}(ik_\zeta\hat{g})\mathcal{T}^{-1}(ik_\psi\hat{\chi})\right]\Big),
\end{aligned} \tag{2.263}$$

$$\mathrm{IV} = +\frac{\mathbf{b}}{m}\cdot(\mu\nabla B + \nabla\mathcal{E}_\Omega)\frac{\partial f}{\partial v_\parallel} \to v_R\left(\mu B \mathcal{G} + \frac{1}{2}\frac{\partial \mathcal{E}_R}{\partial s}\mathcal{F}\right)\frac{\partial \hat{f}}{\partial v_\parallel}, \tag{2.264}$$

$$\begin{aligned}
\mathrm{V} &= -\mathbf{v}_\chi\cdot\nabla F_M \to \mathrm{i}k_\alpha\hat{\chi}\mathcal{E}^{\alpha\psi}\left[\frac{1}{L_n} + \frac{m_R\Omega^2}{T_R}\mathcal{L} + E_T\frac{1}{L_T} + \frac{2v_\parallel}{v_R}\frac{RB_t}{B}u'\right]F_M\\
&\quad +\rho_*\frac{\partial \hat{\chi}}{\partial s}\mathcal{E}^{s\psi}\left[\frac{1}{L_n} + \frac{m_R\Omega^2}{T_R}\mathcal{L} + E_T\frac{1}{L_T} + \frac{2v_\parallel}{v_R}\frac{RB_t}{B}u'\right]F_M,
\end{aligned} \tag{2.265}$$

$$\begin{aligned}
\mathrm{VI} &= -\mathbf{v}_D\cdot\nabla F_M \to \frac{1}{Z}\left[T_R E_D \mathcal{D}^\psi + 2 m_R v_R v_\parallel \Omega \mathcal{H}^\psi + m_R\Omega^2 I^\psi + Z\mathcal{E}^{s\psi}\frac{\partial \Phi}{\partial s}\right]\\
&\quad \times\left[\frac{1}{L_n} + \frac{m_R\Omega^2}{T_R}\mathcal{L} + E_T\frac{1}{L_T} + \frac{2v_\parallel}{v_R}\frac{RB_t}{B}u'\right]F_M,
\end{aligned} \tag{2.266}$$

$$\mathrm{VII} = -\frac{Ze}{T}v_\parallel\mathbf{b}\cdot\nabla\langle\phi\rangle F_M \to -\frac{Z}{T_R}v_R v_\parallel \mathcal{F}\frac{\partial \widehat{\langle\phi\rangle}}{\partial s}F_M, \tag{2.267}$$

$$\begin{aligned}
\mathrm{VIII} &= -\frac{Ze}{T}\mathbf{v}_D\cdot\nabla\langle\phi\rangle F_M \to\\
&\quad -\mathrm{i}\left[E_D\mathcal{D}^\alpha + \beta' v_\parallel^2\mathcal{E}^{\psi\alpha} + \frac{2m_R v_R}{T_R}v_\parallel\Omega\mathcal{H}^\alpha + \frac{m_R\Omega^2}{T_R}\mathcal{I}^\alpha + \frac{Z}{T_R}\mathcal{E}^{\beta\alpha}\frac{\partial \Phi}{\partial x_\beta}\right]k_\alpha\,\widehat{\langle\phi\rangle}\,F_M + \\
&\quad -\rho_*\left[E_D\mathcal{D}^s + \beta' v_\parallel^2\mathcal{E}^{\psi s} + \frac{2m_R v_R}{T_R}v_\parallel\Omega\mathcal{H}^s + \frac{m_R\Omega^2}{T_R}\mathcal{I}^s + \frac{Z}{T_R}\mathcal{E}^{\beta s}\frac{\partial \Phi}{\partial x_\beta}\right]\frac{\partial \widehat{\langle\phi\rangle}}{\partial s}F_M
\end{aligned} \tag{2.268}$$

where the arrows represent the transformations to normalised ($\rightharpoonup$) and Fourier ($\rightarrow$) quantities and where

$$\hat{\chi} = \widehat{\langle\phi\rangle} - 2v_R v_\parallel \widehat{\langle A_\parallel \rangle}, \tag{2.269}$$

and

$$\hat{g} = \hat{f} + \frac{2Z}{T_{\rm R}} v_{\rm R} v_\parallel \widehat{\langle A_\parallel \rangle} F_M, \tag{2.270}$$

$$E_D = v_\parallel^2 + \mu B, \qquad E_T = v_\parallel^2 + 2\mu B + \mathcal{E}_R - \frac{3}{2}. \tag{2.271}$$

The equations above apply to each of the species individually.

In presence of magnetic field compression $\hat{\chi}$ becomes

$$\hat{\chi} = \widehat{\langle\phi\rangle} + \frac{2\mu T_R}{Z} \widehat{\langle B_{1\parallel}\rangle} - 2v_R v_\parallel \widehat{\langle A_\parallel \rangle} \tag{2.272}$$

and there are two additional linear terms which take the form

$$\begin{aligned} \text{X} &= -\frac{F_M}{T} v_\parallel \mathbf{b} \cdot \left(\mu \nabla \langle B_{1\parallel}\rangle\right) \;\rightarrow\; -2v_R v_\parallel \mu F_M \mathcal{F} \frac{\partial \widehat{\langle B_{1\parallel}\rangle}}{\partial s} & (2.273) \\ \text{XI} &= -\frac{F_M}{T} \mathbf{v}_D \cdot \left(\mu \nabla \langle B_{1\parallel}\rangle\right) \;\rightarrow\; -\frac{i}{Z} F_M 2T_R \mu \left(E_D \mathcal{D}^\beta + v_\parallel^2 \beta' \mathcal{E}^{\psi\beta}\right) k_\beta \widehat{\langle B_{1\parallel}\rangle} & (2.274) \\ & \qquad - \rho_* \frac{1}{Z} F_M 2T_R \mu \left(E_D \mathcal{D}^s + v_\parallel^2 \beta' \mathcal{E}^{\psi s}\right) k_\beta \widehat{\langle B_{1\parallel}\rangle} \end{aligned}$$

The only nonlinearity in the equations is the $\mathbf{v}_\chi \cdot \nabla g$ term (term III, Eq. (2.263)). Linear stability can be investigated by removing this term. In the linear case the bi-normal modes are decoupled, but different radial modes can still be coupled over the parallel boundary conditions. An unstable mode will grow to infinite amplitude, and GKW therefore applies a scaling of the equations at every time-step to obtain a stationary solution. The nonlinear term (III) is implemented using FFTs. The velocity $\mathbf{v}_\chi$ and ∇g are calculated in Fourier space, then transformed to real space and multiplied. The product is transformed back into Fourier space.

The fields ϕ and $A_\parallel$ have to be obtained through a solution of the Poisson equation and Ampere's law. These can be found in the literature [33]. Using the normalisations of GKW the Poisson equation can be written in the form

$$\sum_{sp} Z_{sp} n_{R_0,sp} \left[2\pi B \int {\rm d}v_\parallel {\rm d}\mu J_0(k_\perp \rho_{sp}) \hat{g}_{sp} + \frac{Z_{sp}}{T_{Rsp}} [\Gamma(b_{sp}) - 1] \exp(-\mathcal{E}_{R{\rm sp}}) \hat{\phi} \right] = 0, \tag{2.275}$$

where

$$b = \frac{1}{2} m_R T_R (k_\perp \rho_* R_{\rm ref} / Z B^2)^2 = \overbrace{\frac{1}{2} \frac{k_\perp^2 m^2 v_{\rm th}^2}{e^2 B^2}}^{\text{in original units}} . \tag{2.276}$$

This equation differs from the literature [33] only in the polarisation term which includes the rotational density variation in the Maxwellian [51]. An often employed approximation for the electron dynamics is the so called adiabatic response. In the adiabatic electron limit only the ion dynamics is simulated and the Poisson equation above is evaluated as

$$\sum_{sp,ions} Z_{sp} n_{R_0,sp} \left[2\pi B \int {\rm d}v_\parallel {\rm d}\mu J_0(k_\perp \rho_{sp}) \hat{g}_{sp} + \frac{Z_{sp}}{T_{Rsp}} [\Gamma(b_{sp}) - 1] \exp(-\mathcal{E}_{R{\rm sp}}) \hat{\phi} \right] = \frac{n_{R_0,e} \exp(-\mathcal{E}_{R{\rm e}})}{T_{Re}} (\hat{\phi} - \widehat{\{\phi\}}), \tag{2.277}$$

where the index e refers to the electrons, and the sum on the left hand side of the equation is over all ion species. Further details on the evaluation of this equation are given in Section 8.7.

The vector potential must be calculated from Ampere's law

$$\left(k_\perp^2 + \beta \sum_{sp} \frac{Z_{sp}^2 n_{R_0,sp}}{m_{Rsp}} \exp(\mathcal{E}_{R\mathrm{sp}})\Gamma(b_{sp}) \right) \hat{A}_\parallel = \beta \sum_{sp} Z_{sp} v_{Rsp} n_{R_0,sp} \times 2\pi B \int \mathrm{d}v_\parallel \int \mathrm{d}\mu v_\parallel J_0 \hat{g}_{sp}. \tag{2.278}$$

Note that both field equations are formulated for $\hat{g}$ rather than f. The reason is that it is g which is integrated forward in time. Only after the fields are calculated for the new time point, can f be reconstructed. The Maxwell correction in g leads to the second term in the brackets on the left hand side of the equation above. This term can lead to a cancellation problem, and the integral over the Maxwellian is therefore performed numerically in a similar manner as the term on the right hand side [15] rather than the analytic form which is given above.

The field equations in the fully electromagnetic case can be found in section 1.4.1.

Chapter 3

Collisions

The previous sections have neglected the collision operator. In this section we describe the collisions as implemented in GKW. The present implementation neglects the finite Larmor radius effects. The collisions enter the evolution equation as an additional term

$$\frac{\partial \hat{f}_a}{\partial t} = C(\hat{f}_a). \tag{3.1}$$

Here we have added the index a to denote the species. Useful expressions for the collision operator (C) have been published in the literature, usually in the (v, θ) coordinate system, where v is the velocity and θ is the pitch angle. Since the perturbed distribution function is small $f \approx \rho_* F_M$, GKW uses a linearised collision operator with the Maxwellian as background. This operator has the following form [46]

$$C(\hat{f}_a) = \sum_b \frac{1}{v^2}\frac{\partial}{\partial v}\left[v^2\left(D_{vv}^{a/b}\frac{\partial \hat{f}_a}{\partial v} - F_v^{a/b}\hat{f}_a\right)\right] + \frac{1}{v\sin\theta}\frac{\partial}{\partial\theta}\left[\sin\theta D_{\theta\theta}^{a/b}\frac{1}{v}\frac{\partial \hat{f}_a}{\partial\theta}\right]. \tag{3.2}$$

Here the sum is over all species b. $D_{\theta\theta}$ represents the pitch angle scattering, while D_{vv} is the energy scattering and F_v is the slowing down force. For convenience we split this collision operator in three parts

$$C(\hat{f}) = C_\theta(\hat{f}) + C_{vv}(\hat{f}) + C_v(\hat{f}), \tag{3.3}$$

where the first part is due to $D_{\theta\theta}$, the second due to D_{vv} and the third due to F_v. The normalisation of the collision operator follows from the normalisation of the gyrokinetic equation as set out in section 2.1, giving $C_N(\hat{f}_a) = \frac{R_{\rm ref} v_{tha}^3}{n_a \rho_* v_{\rm thref}} C(\hat{f}_a)$ with the diffusion and friction force coefficients made dimensionless accordingly.

The coefficients can be obtained from the literature [46] (and the references cited therein) and written in normalised form, suppressing the subscript N for the normalised quantities:

$$D_{\theta\theta}^a = \sum_b \frac{\Gamma^{a/b}}{4v_a}\left[\left(2 - \frac{1}{v_b^2}\right)\mathrm{erf}(v_b) + \frac{1}{v_b}\mathrm{erf}'(v_b)\right], \tag{3.4}$$

$$D_{vv}^a = \sum_b \frac{\Gamma^{a/b}}{2v_a}\left[\frac{1}{v_b^2}\mathrm{erf}(v_b) - \frac{1}{v_b}\mathrm{erf}'(v_b)\right], \tag{3.5}$$

$$F_v^a = -\sum_b \frac{\Gamma^{a/b}}{v_a^2}\frac{m_{Ra}}{m_{Rb}}[\mathrm{erf}(v_b) - v_b \mathrm{erf}'(v_b)]. \tag{3.6}$$

where erf is the standard definition of the error function. The normalised velocity in these equations is defined as

$$v_\gamma = \frac{v_{Ra}}{v_{R\gamma}}[v_\parallel^2 + 2\mu B]^{1/2}, \tag{3.7}$$

with $\gamma = a$ or $\gamma = b$. The error function is

$$\mathrm{erf}(u) = \frac{2}{\sqrt{\pi}} \int_0^u \exp[-x^2]\mathrm{d}x \tag{3.8}$$

$$\mathrm{erf}'(u) = \frac{2}{\sqrt{\pi}} \exp[-u^2] \tag{3.9}$$

Finally the constant $\Gamma^{a/b}$ is given by

$$\Gamma^{a/b} = \frac{R_{\mathrm{ref}} n_b Z_a^2 Z_b^2 e^4 \ln \Lambda^{a/b}}{4\pi\epsilon_0^2 m_a^2 v_{tha}^4} = 6.5141 \cdot 10^{-5} \frac{R_{\mathrm{ref}} n_{\mathrm{ref}}^{19}}{(T_{\mathrm{ref}}^k)^2} \frac{n_{Rb} Z_a^2 Z_b^2 \ln \Lambda^{a/b}}{T_{Ra}^2}, \tag{3.10}$$

where R_{ref} must be given in meters, n_{ref}^{19} is the reference density in units of $10^{19}\ m^{-3}$, and T_{ref}^k is the reference temperature in units keV. The Coulomb logarithm is calculated as

$$\begin{aligned}
\ln \Lambda^{e/e} &= 14.9 - \frac{1}{2}\ln[0.1 n_{\mathrm{R}}^e n_{\mathrm{ref}}^{19}] + \ln[T_{\mathrm{R}}^e T_{\mathrm{ref}}^k] && (3.11)\\
\ln \Lambda^{e/i} &= 17.2 - \frac{1}{2}\ln[0.1 Z^2 n_{\mathrm{R}}^e n_{\mathrm{ref}}^{19}] + 1.5\ln[T_{\mathrm{R}}^e T_{\mathrm{ref}}^k] \qquad \text{if } T_e < 10Z^2\,\mathrm{eV} && (3.12)\\
\ln \Lambda^{e/i} &= 14.8 - \frac{1}{2}\ln[0.1 n_{\mathrm{R}}^e n_{\mathrm{ref}}^{19}] + \ln[T_{\mathrm{R}}^e T_{\mathrm{ref}}^k] \qquad \text{if } T_e \geq 10Z^2\,\mathrm{eV} && (3.13)\\
\ln \Lambda^{i/e} &= \ln \Lambda^{e/i} && (3.14)\\
\ln \Lambda^{i_1/i_2} &= 17.3 - \ln\left[\frac{Z_1 Z_2 (m_1 + m_2)}{m_1 T_{\mathrm{R}2} T_{\mathrm{ref}}^k + m_2 T_{\mathrm{R}1} T_{\mathrm{ref}}^k}\right] - \frac{1}{2}\ln\left[0.1\frac{n_{\mathrm{ref}}^{19}}{T_{\mathrm{ref}}^k}\right] - \frac{1}{2}\ln\left[\frac{n_{\mathrm{R}1} Z_1^2}{T_{\mathrm{R}1}} + \frac{n_{\mathrm{R}2} Z_2^2}{T_{\mathrm{R}2}}\right] && (3.15)
\end{aligned}$$

where the superscripts e and i stand for electron and ion, respectively.

Note that the reference density and temperature are defined independently for the collisions and do not need to be equal to the reference density and temperature used in Sec. 2.1 for the species parameters. However, the electron density entering the calculation of the Coulomb logarithm is taken to be $n_{\mathrm{R}}^e n_{\mathrm{ref}}^{19}$ and therefore depends on the normalised electron density given as in an input in the species namelist. The same comments pertains for the temperatures entering the calculation of the Coulomb logarithm.

Using the relation between (v, θ) and the normalised coordinates $(v_\parallel, \mu)$ used in GKW, and performing the coordinate transform, one arrives at the explicit expressions given below. The pitch angle scattering term is

$$\begin{aligned}
C_\theta \quad \rightarrow \quad & 2v_R \frac{\partial}{\partial v_\parallel}\left[\frac{\mu D_{\theta\theta}}{v_\parallel^2 + 2\mu B}\left(B\frac{\partial \hat{f}}{\partial v_\parallel} - v_\parallel \frac{\partial \hat{f}}{\partial \mu}\right)\right] \\
+ \quad & \frac{2v_R}{B}\frac{\partial}{\partial \mu}\left[\frac{v_\parallel \mu D_{\theta\theta}}{v_\parallel^2 + 2\mu B}\left(-B\frac{\partial \hat{f}}{\partial v_\parallel} + v_\parallel \frac{\partial \hat{f}}{\partial \mu}\right)\right].
\end{aligned} \tag{3.16}$$

The energy scattering yields

$$\begin{aligned}
C_{vv} \quad \rightarrow \quad & v_R \frac{\partial}{\partial v_\parallel}\left[\frac{v_\parallel D_{vv}}{v_\parallel^2 + 2\mu B}\left(v_\parallel\frac{\partial \hat{f}}{\partial v_\parallel} + 2\mu \frac{\partial \hat{f}}{\partial \mu}\right)\right] \\
+ \quad & v_R \frac{\partial}{\partial \mu}\left[\frac{2\mu D_{vv}}{v_\parallel^2 + 2\mu B}\left(v_\parallel\frac{\partial \hat{f}}{\partial v_\parallel} + 2\mu \frac{\partial \hat{f}}{\partial \mu}\right)\right].
\end{aligned} \tag{3.17}$$

The slowing down can be written as

$$C_v \quad \rightarrow \quad -v_R \frac{\partial}{\partial v_\parallel}\left[\frac{v_\parallel F_v}{[v_\parallel^2 + 2\mu B]^{1/2}}\hat{f}\right]$$

$$-v_R \frac{\partial}{\partial \mu}\left[\frac{2\mu F_v}{[v_\parallel^2 + 2\mu B]^{1/2}}\hat{f}\right]. \tag{3.18}$$

The linearised form of the Fokker-Planck collision operator above conserves particle number but does not conserve like particle momentum or energy. The conservation of parallel momentum and energy can be reintroduced using the following simplified model. We are interested in momentum conservation for like particle collisions. The change in parallel momentum due to the collision operator above is

$$2\pi B \int \mathrm{d}v_\parallel \int \mathrm{d}\mu v_\parallel C(\hat{f}) \tag{3.19}$$

This loss of momentum can be corrected for by adding an ad hoc correction term

$$\frac{\partial \hat{f}}{\partial t} = C_{mom} v_\parallel F_M, \tag{3.20}$$

but the energy term isn't simply

$$\frac{\partial \hat{f}}{\partial t} = C_{ene}(v_\parallel^2 + 2\mu B) F_M \tag{3.21}$$

as this would have the effect of adding particles, conservation of which is accounted for by the differencing and boundary conditions. Therefore we use

$$\frac{\partial \hat{f}}{\partial t} = C_{ene}\left((v_\parallel^2 + 2\mu B) - A\right) F_M \tag{3.22}$$

where A is calculated by taking the zeroth moment of the equation

$$\frac{\partial \hat{f}}{\partial t} = C(\hat{f}) + C_{ene}\left((v_\parallel^2 + 2\mu B) - A\right) F_M \tag{3.23}$$

and setting the change in density to zero. Since particle conservation is determined by the differencing and boundary conditions, then the coefficient A turns out to be simply:

$$A = \frac{\int \mathrm{d}v_\parallel \int \mathrm{d}\mu v^2 F_M}{\int \mathrm{d}v_\parallel \int \mathrm{d}\mu F_M} \tag{3.24}$$

which analytically can be shown to be $\frac{3T}{2}$, but in GKW is calculated numerically to ensure highest precision conservation.

The constant C_{mom} then follows from working out the momentum change due to this term.

$$2\pi B C_{mom} \int \mathrm{d}v_\parallel \mathrm{d}\mu v_\parallel^2 F_M + 2\pi B \int \mathrm{d}v_\parallel \int \mathrm{d}\mu v_\parallel C(\hat{f}) = 0. \tag{3.25}$$

while the constant C_{ene} is determined by taking the energy moment $\frac{mv^2}{2}$,

$$2\pi B C_{ene} \int \mathrm{d}v_\parallel \mathrm{d}\mu v^2 (v^2 - A) F_M + 2\pi B \int \mathrm{d}v_\parallel \int \mathrm{d}\mu (v_\parallel^2 + 2\mu B) C(\hat{f}) = 0. \tag{3.26}$$

here we have suppressed the $m/2$ factors as they cancel.

The present form of the collision operator drives the distribution towards a Maxwellian with the specified temperature. In the local problem one specifies the temperature as an input parameter and no evolution in the temperature is retained. A collision operator that maintains this temperature can therefore be considered acceptable. The steady state distribution function for various velocity space resolutions is plotted in Fig.3.3.

To switch on the conservation terms, which can be switched on or off independently, the parameters *mom_conservation* = `.true.` and *ene_conservation* = `.true.`. However it must be noted that for energy conservation to machine accuracy, the parameter *mass_conserve* = `.true.` must also be set, to enforce conservation at the velocity boundaries (although this may not be the most physical boundary condition).

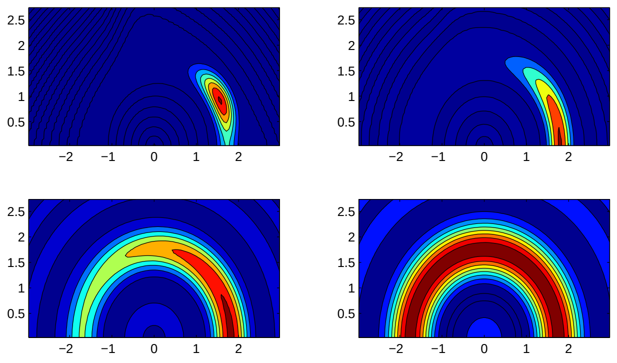


Figure 3.1: Snapshot of velocity space as three different time intervals showing the effect of pitch angle scattering on the velocity space. Coordinates are $v_\parallel$ horizontally and $v_\perp$ vertically. All data shown in this chapter was run with 100 parallel velocity points and 50 μ points non-equally spaced.

In GKW the μ-direction grid is non-uniform which represents a problem when differencing the collision terms, however it is seen that while non-uniform in μ it is uniform in the $v_\perp$ coordinate ($mu = v_\perp^2/2B$) and as such the collision operator is solved using this as the coordinate as this ensures machine accurate flux conservation and momentum conservation. To get the collision operator in this form the following simple transformation must be performed:

$$\begin{aligned} 2\mu B &= v_\perp^2 && (3.27)\\ 2\mu\partial/\partial\mu &= \partial/\partial v_\perp . && (3.28)\end{aligned}$$

3.1 Collisions tutorial

In the previous section we derived the form of the three terms of the Fokker-Plank operator. Here we will show, using data produced by GKW and our implementation of the previous to show what effect the operator has on the velocity space structure.

When running GKW, collisions are included if *collisions* = `.true.` in the CONTROL namelist

3.1.1 Pitch-angle scattering

This effect is switched on by setting *pitch_angle* = `.true.` in the collisions input deck. Pitch-angle scattering conserves momentum and therefore will have the effect of isotropising the distribution function along arcs of constant velocity $v = \sqrt{v_\parallel^2 + 2\mu B}$.

If we initialise the code with narrow, shifted gaussian in velocity space, then as time progresses then pitch-angle scattering will have the effect of producing a ring in velocity space (As seen in figure 3.1).

3.1.2 Energy scattering

This effect is switched on by setting *en_scatter* = *.true.* in the collisions input deck. The energy scattering term spreads the distribution function in energy.

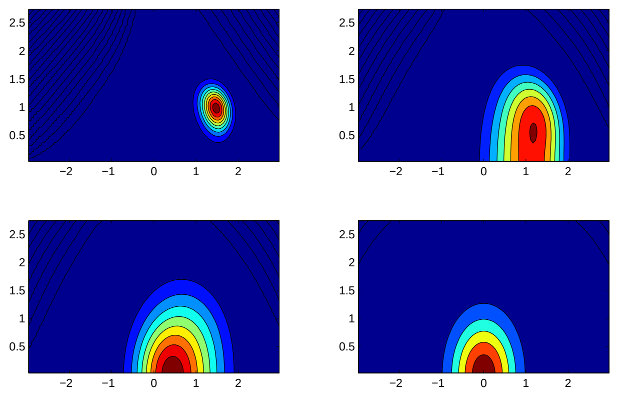


Figure 3.2: Snapshot of velocity space as four different time intervals showing the effect of all the terms in the collision operator on the velocity space. The final state is a Maxwellian centred on zero.

3.1.3 Collisional friction

This effect is switched on by setting *friction_coll* = *.true.* in the collisions input deck. The collisional friction term advects the distribution function towards the origin in velocity space.

When you combine all three effects we get an evolution of the distribution function towards a Maxwellian centred on zero and with a width of a single normalised thermal velocity.

3.2 Velocity space boundary conditions

There are currently two options for the boundary conditions in velocity space. The input deck parameter *mass_conserve* changes between open boundary conditions (when set to `.false.`, default) and zero flux boundary conditions (when set to `.true.`). The former is more physical, but the latter is useful for testing of conservation properties to machine precision. The difference in mass conservation between the two is very small.

3.3 User defined collision-frequencies

The collision frequency can be defined in two ways. Firstly by specifying *nref* which is the particle density in units of 10^{19} particles per m^3, *tref*, the temperature in keV and *rref* which is the toroidal major radius. These parameters are what the corresponding parameters are normalised to in the code and are never usually specified. The normalised collision frequency $\Gamma^{a/b} = \nu_{ab} R_{\rm ref}/v_{\rm thref}$ is given by,

$$\Gamma_N^{a/b} = \frac{R_{\rm ref} n_b Z_a^2 Z_b^2 e^4 \ln \Lambda^{a/b}}{4\pi\epsilon_0^2 m_a^2 v_{tha}^4} = 6.5141 \cdot 10^{-5} \frac{R_{\rm ref} n_{\rm ref}^{19}}{(T_{\rm ref}^k)^2} \frac{n_{Rb} Z_a^2 Z_b^2 Z_{\rm eff}^{a/b} \ln \Lambda^{a/b}}{T_{Ra}^2} \tag{3.29}$$

where $Z_{\rm eff}^{a/b} = 1.0$ unless $b = i$ (main ion species) as determined by the species density and $a = e$ as determined by the charge on the species. The value of $Z_{\rm eff}^{e/i}$ is adjusted by the parameter *zeff* in the collisions namelist which is 1.0 by default. Note that if impurity species are included kinetically, then the scaling factor will be automatically modified to be

$$Z_{\rm eff}^{e/i} = 1 + (\mathit{zeff} - Z_{\rm eff}^{\rm sp}) \frac{n_e}{n_i Z_i^2} \tag{3.30}$$

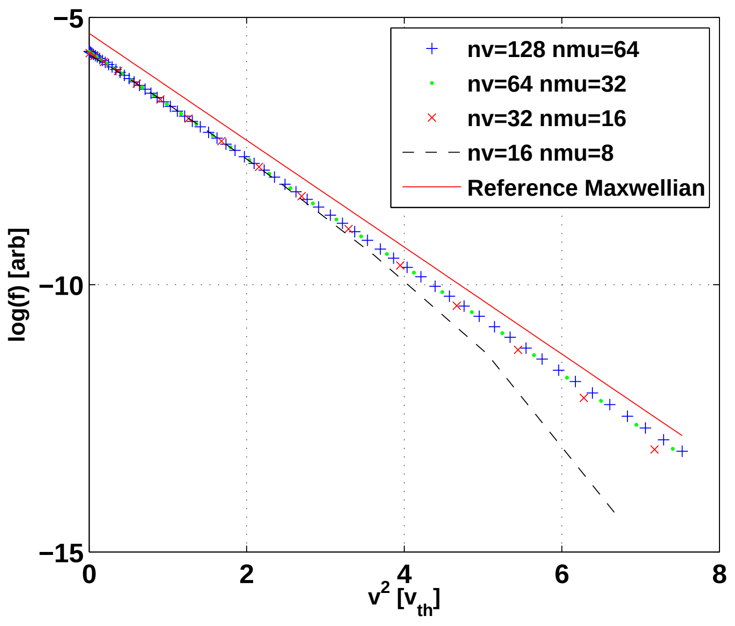


Figure 3.3: Figure showing the logarithm of the distribution function as a function of v^2 for varying velocity space resolutions. A reference Maxwellian is also plotted for comparison.

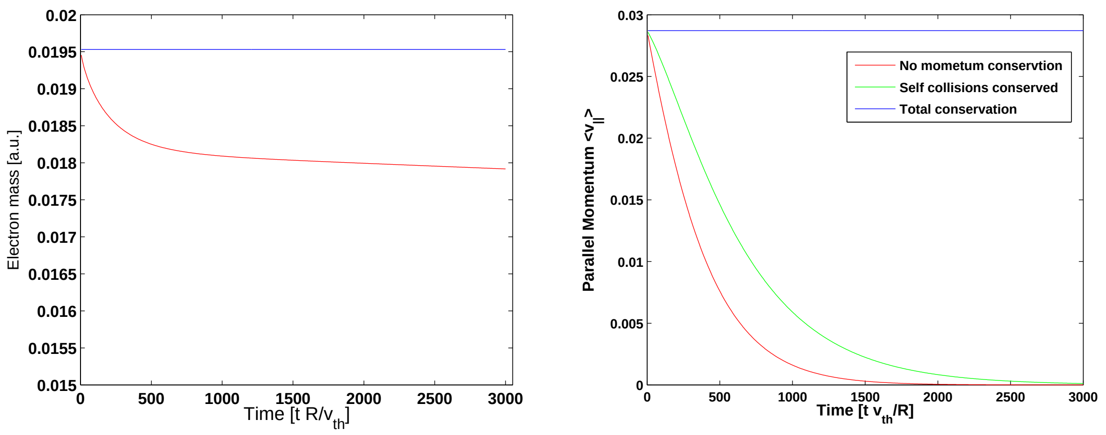


Figure 3.4: (left) Figure shows the electron particle mass for when (blue) flux conserving boundaries and (red) open boundaries are used. When the former is chosen the particle number is conserved to machine precision. Note the y-axis scale. (Right) Shows the parallel momentum for runs with the same initial conditions for (blue) fully conservative (green) self collisions conservative and (red) no conservation at all. Again, the momentum is conserved to machine precision.

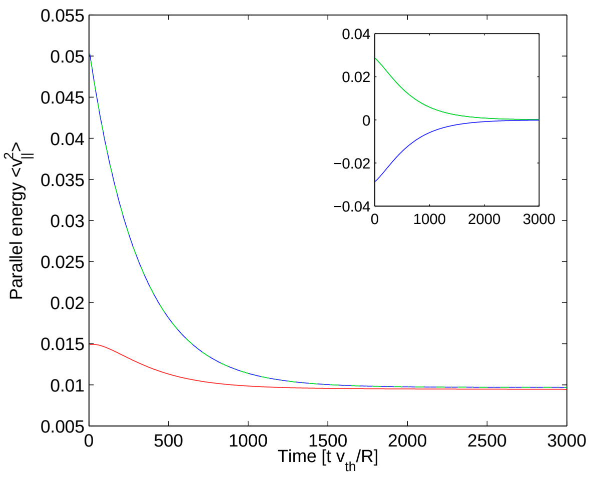


Figure 3.5: Figure shows the energy in the electron distribution for three different initial conditions (green and blue had the same initial width but with a different parallel velocity sign, red was gaussian centered different initial parallel velocity). All three converge on the same value. (Inlay is the parallel momentum)

with $Z_{\text{eff}}^{\text{sp}} = \frac{1}{n_e}\sum_s n_s Z_s^2$ and the sum on 's' performed on all ion species

However if *freq_override* = `.true.` then,

$$\Gamma_N^{a/b} = Z_a^2 Z_b^2 \nu_{\text{ref}} \frac{n_{Rb}}{T_{Ra}^2} \frac{\ln \Lambda^{a/b}}{\ln \Lambda^{i/i}} \tag{3.31}$$

Where ν_{ref} is the user defined frequency defined as *coll_freq* in the input deck. This is assumed to be the singly charged ion - ion (proton or deuteron) collision frequency. To get agreement with the first method this can be set to $6.5141 \cdot 10^{-5} R_{\text{ref}} n_{\text{ref}}^{19} \ln \Lambda^{i/i}/(T_{\text{ref}}^k)^2$. This value is then scaled to all the other species respectively. For this to work the first species in the input deck should be singly charged ions.

3.4 Direct benchmarks

Here outlined is a direct analytical benchmark of the collision operator. The frictional slowing down of a particle can be calculated analytically, and accuratly approximated in the code by using a very narrow, high energy gaussian in velocity space. The velocity of this pulse can then be tracked and compared with the analytic expression.

The slowing down time of a particle can be written as:

$$\tau^s = -\frac{U}{\frac{\partial U}{\partial t}} \tag{3.32}$$

where $\tau^s = 1/\nu_S$, which is defined as (we consider only ion-ion collisions):

$$\nu_S^{a/b} = -\left[(1 + \frac{m_a}{m_b})\psi(x)\right] \nu_0^{a/b} \tag{3.33}$$

where $\nu_0^{a/b} = \Gamma_N^{a/b} = \frac{R_{\text{ref}} n_b Z_a^2 Z_b^2 e^4 \ln \Lambda^{a/b}}{4\pi\epsilon_0^2 m_a^2 v_a^3}$, where $\psi(x)$ is the Maxwell integral and can be shown to be $\psi(x) = erf(x) - x\frac{\partial erf(x)}{\partial x}$. and x is defined as v^a/v_{th}^b. This ordinary differential equation can then be solved using standard methods. The comparison between GKW and this result is shown in Fig. 3.6. Agreement between the two curves is very good, implying that the friction and energy scatter terms are operating correctly in the current implementation. A benchmark of the pitch angle scattering against another code is presented in Sec. 11.1.

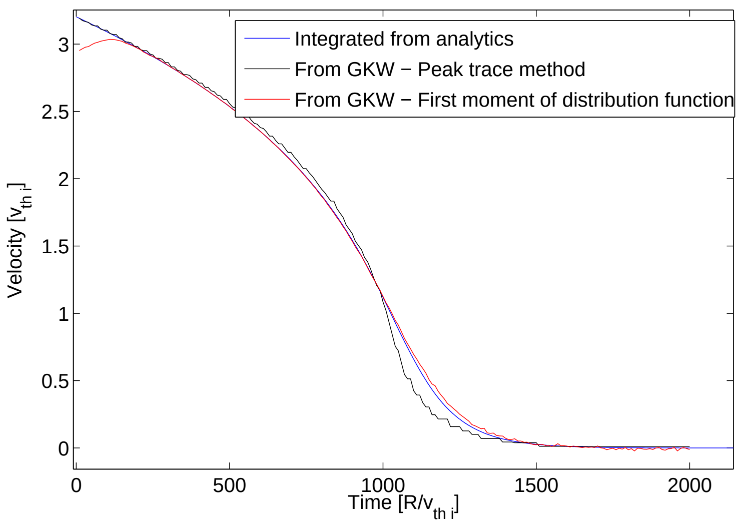


Figure 3.6: Measurement of the velocity of a highly localised pulse in velocity space as a function of time. The result from GKW is shown utilising two methods, one is the first moment of the distribution function, while the other is simply a trace of the spike. The blue line is the solution of the ODE described above.

Chapter 4

Rotation and poloidal asymmetries

The co-moving system employed in GKW corresponds to a frame that rotates as a rigid body with constant frequency Ω. The Coriolis drift, centrifugal drift, centrifugal trapping and centrifugal potential due to the rotation of the plasma are all incorporated in the complete set of equations described above. The possible radial gradient in the toroidal rotation is treated through a radial gradient in the averaged parallel velocity u' of the background. The radial gradient of the perpendicular velocity ($E \times B$ shear flow) has so far been ignored, but is known to play an important role in the saturation of the turbulence.

In this chapter we describe how the centrifugal potential is calculated from the species inputs, and how the the $E \times B$ shearing is treated in GKW. Much of this Chapter appears in greater detail in Ref. [51].

4.1 Centrifugal potential

We recall that the rotation of the plasma leads to an equilibrium density variation within a flux surface

$$n(s) = n_{R_0,s} \exp\left(\frac{-Z_s\langle\Phi\rangle}{T_{R,s}} + \frac{m_s\Omega^2(R^2 - R_0^2)}{T_{R,s}}\right) \tag{4.1}$$

(normalised units). The background equilibrium potential Φ in the rotating frame is found by applying the quasi-neutrality condition over all the species

$$Q(\Phi) = \sum_s Z_s n_{R_0,s} \exp\left(\frac{-Z_s\langle\Phi\rangle}{T_{R,s}}\right) \exp\left(\frac{m_s\Omega^2(R^2 - R_0^2)}{T_{R,s}}\right) = 0 \tag{4.2}$$

Since there is only one negative species and the exponential function is monotonically increasing, the equation $Q(\Phi) = 0$ will always have exactly one root. In GKW, Φ is calculated numerically for arbitrary species combinations from the quasi-neutrality condition by an iterative root finding bisection algorithm. The calculation is done once for each point in s during the initialisation phase of the code. The terms II, VI, and VIII (2.262, 2.266, 2.268) also require the ψ and s derivatives of Φ, which are calculated using a fourth-order finite difference.

The bisection algorithm begins with an upper limit estimate Φ_a for Φ. The initial value used in GKW is (somewhat arbitrarily) taken as

$$\left[\max\{\log(1 + |Z_s n_s|)\} + \max\{m_s \log(1 + n_s)\}\right]\Omega^2(R^2 - R_0^2) + 0.1 \tag{4.3}$$

which should always contain the root, but be small enough to prevent exponentiation overflows. It is possible that extreme species input data could break the solver, and the upper limit estimate might need to be adjusted for those cases.

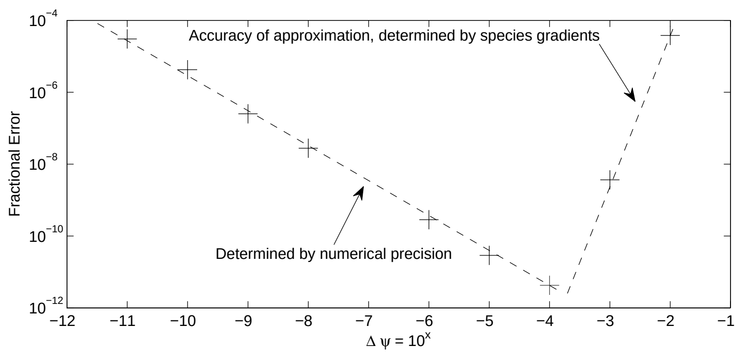


Figure 4.1: Numerical experiment to determine optimum $\Delta\psi$ for calculating the radial derivative of the centrifugal potential Φ for the singly charged ions Waltz standard case. The experiment was conducted in double precision and the fractional error is calculated against the analytic result Eq. 4.5.

The starting lower limit is estimated as $\Phi_b = -\Phi_a$. The value is chosen to ensure that the root lies in the range (Φ_b, Φ_a). In each step the mid value $\Phi_{\text{est}} = (\Phi_a + \Phi_b)/2$ is taken and the value of $Q(\Phi_{\text{est}})$ is calculated. If $Q(\Phi_{\text{est}}) > 0$, then the upper estimate is updated ($\Phi_a = \Phi_{\text{est}}$), and if $Q(\Phi_{\text{est}}) < 0$, then the lower estimate is updated ($\Phi_b = \Phi_{\text{est}}$). The process repeats until the solution has converged to within machine accuracy. The bisection algorithm, while not the fastest possible, is guaranteed to converge as long as the bounds are appropriate and the function Q has only one root in the initial range. In practice, the convergence occurs in about 55 iterations using the initial estimates specified above. The function $Q(\Phi, s, \psi)$ is evaluated in module *components* as *cf_quasineutral* called from the bisection algorithm *centrifugal_energy* in module *rotation*.

In the local flux tube model, the gyrokinetic equation is solved only on one flux surface (i.e. at one position in ψ), keeping radial derivatives of equilibrium quantities. To consistently calculate the radial derivative of Φ, equation 4.2 is also solved numerically on adjacent flux surfaces using $Q(\Phi, s, \psi + \Delta\psi)$, using the radial derivatives of the species quantities to calculate the temperature and densities on adjacent flux surfaces.

The choice of $\Delta\psi$ used in the code is motivated by a compromise between machine precision when dividing by small numbers, and the accuracy of the finite difference as $\Delta\psi \to 0$. The discrepancy between the root finding algorithm and the analytic solution (Eq. 4.5) is plotted in figure 4.1. The value of $\Delta\psi = 1e-4$ is implemented as a permanent choice, which for the case investigated minimises the error to less than 10^{-11} in double precision.

In the case of a plasma of only singly charged ions, Eq. 4.2 solves exactly to give

$$\Phi = \underbrace{\frac{T_e T_i}{T_e + T_i}\left(\frac{m_i}{T_i} - \frac{m_e}{T_e}\right)}_{\textit{cf\_mass\_weight}} \Omega^2 (R^2 - R_0^2). \tag{4.4}$$

with radial derivative

$$\frac{\partial \Phi}{\partial \psi} = \underbrace{\left[\frac{m_i \frac{\partial T_e}{\partial \psi} - m_e \frac{\partial T_i}{\partial \psi}}{T_e + T_i} + \frac{m_e T_i - m_i T_e}{(T_e + T_i)^2}\left(\frac{\partial T_e}{\partial \psi} + \frac{\partial T_i}{\partial \psi}\right)\right]}_{\textit{cf\_dtf1}} \Omega^2 (R^2 - R_0^2) + 2\Omega^2 \underbrace{\left[\frac{m_i T_e - m_e T_i}{T_e + T_i}\right]}_{\textit{cf\_mass\_weight}} \left(R\frac{\partial R}{\partial \psi} - R_0 \frac{\partial R_0}{\partial \psi}\right) \tag{4.5}$$

If the bulk ion species is singly charged, these quantities are evaluated in the code in *rotation* with the labelled function provided by *components* to provide a check on the solution of the root finding algorithm.

The density n_R and density gradient $1/L_n = (1/L_n)_{R_0}$ (Eq. 2.8) are defined for each species at the point on the flux surface at which $R = R_0$. Two choices for the definition of R_0 are available in the code, which are selected by the input variable *R0_loc* in the *geom* namelist. The option `R0_loc='axis'` sets R_0 as the major radius of the magnetic axis (but the reference point is not on the axis itself), whilst the default

`R0_loc='LFS'` sets R_0 to be the value of R at $s = 0$, in the plane of the magnetic axis at the low field side (for general geometry this may *not* be equivalent to the maximum R). This choice alters only the definition of the geometry quantities $\mathcal{J}$ and $\mathcal{K}$. In the rotating case, the effective density gradient varies over the flux surface and there is no longer one obvious definition of the density gradient as a dimensionless quantity. For instance

$$\frac{1}{L_n^E} = -\frac{1}{n}\frac{\partial n}{\partial \psi}, \qquad \frac{1}{L_n^{e1}} = -\frac{1}{\{n\}}\frac{\partial n}{\partial \psi}, \qquad \frac{1}{L_n^{e2}} = -\frac{1}{n_{R_0}}\frac{\partial n}{\partial \psi}, \tag{4.6}$$

can each be argued to be a relevant quantity. The most appropriate may be determined by the nature of the particular instability being examined. These definitions can be evaluated to be

$$\frac{1}{L_n^e} = \frac{\partial \mathcal{E}_R}{\partial \psi} + \mathcal{E}_R \frac{1}{L_T} + \frac{1}{L_n}\bigg|_{R_0}, \qquad \frac{1}{L_n^{e1}} = \frac{1}{L_n^e}\frac{\exp(-\mathcal{E}_R)}{\{\exp(-\mathcal{E}_R)\}}, \qquad \frac{1}{L_n^{e2}} = \frac{1}{L_n^e}\exp(-\mathcal{E}_R) \tag{4.7}$$

Since the density gradient is an important parameter for determining the stability of many modes, GKW calculates $1/L_N^E$ and $1/L_n^{e1}$ and n/n_{R_0} for each species as a function along the field, which is written to the file `cfdens.dat`, and as a flux surface average and at the outboard midplane, which is written to screen during the initialisation phase of the code. $1/L_n^{e2}$ can be trivially calculated from the other two. It should be noted that whilst the input $1/L_n|_{R_0}$ must always be the same for every species in order to satisfy quasi-neutrality, the effective $1/L_n^E$ can differ for each species. For typical parameters of a deuterium plasma, the variation in effective density and gradient can be 30% at $\Omega = 1$, increasing rapidly for $\Omega > 1$.

4.2 Background $E \times B$ shear flow

The sheared $E \times B$ velocity leads to a stabilisation of turbulence through the breaking up of eddy structures. This $E \times B$ shearing is always present for a purely toroidally rotating plasma but can, of course, also be related to a sheared poloidal rotation.

The equilibrium $E \times B$ rotation

$$\mathbf{v}_s(\psi) = \frac{\mathbf{b} \times \nabla \bar{\Phi}}{B} \tag{4.8}$$

lies inside the flux surface. The shearing rate in the normalised units is defined to be

$$\gamma_E^N = \frac{1}{2}\rho_*^2 \frac{\partial^2 \bar{\Phi}_N}{\partial \psi^2}, \tag{4.9}$$

where the normalisation $\bar{\Phi} = \rho_* \frac{T_{\rm ref}}{e} \bar{\Phi}_N$ is as for the perturbed potential Eq. (2.4). The factor 1/2 is present due to the definition of the reference thermal velocity. Since GKW treats the $E \times B$ shear independently from the centrifugal effects, we use $\bar{\Phi}$ to distinguish the background potential of the shearing from the background potential of the rotating frame. In physical units the shearing rate is

$$\gamma_E = \frac{v_{\rm thref}}{R_{\rm ref}} \gamma_E^N = \frac{1}{B_{\rm ref}} \frac{\partial^2 \bar{\Phi}}{\partial r^2}, \tag{4.10}$$

where $r = (R_{\rm max} - R_{\rm min})/2$. The GKW shearing rate is assumed radially constant and is defined as a flux function. At the radial centre of the flux tube $\mathbf{v_s}$ is chosen to be zero, with the result that there is no net flow over the domain. In the local limit with the approximations of the '$s - \alpha$' model geometry the shearing rate is then equivalent to the familiar definition [58, 59]:

$$\gamma_E \approx \frac{(RB_p)^2}{B_t} \frac{\partial^2 \bar{\Phi}}{\partial \Psi^2} \approx \frac{dv_s}{dr}. \tag{4.11}$$

The flow velocity is normalised as $v_s^N = v_s/v_{\rm thref}$, and as before the index N is dropped in what follows. The sign convention here is opposite to Eq. (2.8), so $\gamma_E < 0$ corresponds to ∇E_0 radially outwards.

Only the Doppler shift of the background $E\times B$ rotation is kept in the description, i.e. the background $E\times B$ rotation is added as an additional convective term for the perturbed distribution ($\mathbf{v}_s \cdot \nabla g$) in the gyrokinetic equation (7.37), and we neglect the acceleration due to the background potential ($\mathbf{v}_D \cdot \nabla \bar{\phi} F_M$). By omitting the latter term, the flow is implemented in conservative form and provides no drive to the turbulence, so that the stabilising effect may be studied in isolation. If the acceleration term were kept, it would provide a Kelvin-Helmholtz drive to the turbulence. This drive is likely to be small compared to the parallel shear drive (which is kept in term VIII), but could be the subject of future work.

With zero net flow across the flux tube one obtains

$$\mathrm{III.B} = -\mathbf{v}_s \cdot \nabla g \rightharpoonup -\rho_*^2 \frac{\partial \bar{\Phi}}{\partial \psi} \mathcal{E}^{\psi\zeta} \frac{\partial g}{\partial \zeta} = -2\gamma_E \psi \mathcal{E}^{\psi\zeta} \frac{\partial g}{\partial \zeta}. \tag{4.12}$$

Defining $\bar{\gamma}_E = 2\gamma_E \mathcal{E}^{\psi\zeta}$ and taking the Fourier transform, the term can be written as a derivative in Fourier space

$$\mathcal{T}(-\bar{\gamma}_E \psi \frac{\partial g}{\partial \zeta}) = \bar{\gamma}_E k_\zeta \frac{\partial \hat{g}}{\partial k_\psi}. \tag{4.13}$$

The derivative represents a continuous advection (and shearing) in Fourier space in k_ψ.

There are a number of subtleties associated with numerical evaluation of this derivative. A finite difference approximation cannot resolve the fine scale structure in Fourier space. The 'harmonic derivative' [60, 61] is a full convolution over all modes and is equivalent to a pseudo-spectral implementation using FFTs (as was used for the nonlinear term in Eq. (2.263)). Although it correctly implements the $E \times B$ shearing term inside the computational domain, it does not represent a homogeneous shear flow because the flow profile does not satisfy the periodicity of the discrete Fourier transform; any turbulent structure passing through the radial boundary will experience a discontinuity in the flow profile (which in the periodic domain has a sawtooth form). Since the local limit requires that the turbulence has no preference for the boundary, periodicity at the boundary must be preserved. The 'shear-periodic' boundary condition [62, 63, 64] that moves with the mean flow accomplishes this for a finite difference representation in the radial direction, but does not naturally translate to the spectral representation.

In coordinates that move with the flow [65, 66, 67], the 'radial' wavenumbers become time dependent:

$$\psi' = \psi, \qquad \zeta' = \zeta - \psi \bar{\gamma}_E t, \tag{4.14}$$
$$k'_\zeta = k_\zeta, \qquad k'_\psi = k_\psi - k_\zeta \bar{\gamma}_E t. \tag{4.15}$$

In these coordinates, the derivative in Fourier space becomes part of the time derivative.

For a gyrokinetic code, a time dependent wave vector requires the re-evaluation of the linear terms and Bessel functions at every time-step and would be computationally prohibitive. By discretising the time dependence of the wavenumbers as a remapping of the solution vector between the original fixed wavevectors, this expensive re-evaluation can be avoided [68]. Using only the fixed wavevector grid, the advection in Fourier space occurs in jumps only when the boundary periodicity is satisfied for a given mode. Explicitly, the remapping is implemented by tracking the number of times each wavevector has been remapped $i_r(k_\zeta)$, and (for $\bar{\gamma}_E > 0$) remapping the solution vector

$$\hat{g}(k_\psi, k_\zeta, s) \rightarrow \hat{g}(k_\psi - \Delta k_\psi, k_\zeta, s), \tag{4.16}$$

when the inequality

$$\mathrm{Int}(k_\zeta \bar{\gamma}_E t / \delta k_\psi) > i_r(k_\zeta), \tag{4.17}$$

is satisfied. Here, Int is the function which returns the *nearest* integer. At the low k_ψ limit the solution vector is discarded. This means that the method is non-conservative, but since the turbulence is characterised by a peaked spectrum (Fig. 11.1), the losses are negligible if the range of radial wavenumbers is suitably wide. For numerical stability, the remapping must occur at the same time for all points along the flux tube. As the metric tensor $\mathcal{E}^{\psi\zeta}$ is a flux function, see Eq. (2.24), this condition is always satisfied for a constant shear rate along the field line.

The method in GKW differs from the standard spectral methods to model homogeneous shear flows in fluid turbulence of Refs. [65, 66, 67], where the wavenumbers must be recalculated and re-meshing occurs for all wavevectors at the same time. In GKW the method used is the one proposed by Hammett et al. in Ref. [68]. Here the 're-meshing' occurs continually (and at different times for each k_ζ), and the wavenumbers stay on their original fixed grid. The accuracy of the method has been argued to be second order on average [68] and is able to capture the physics effects of a background shear flow whilst allowing the desirable features of the flux tube model to be retained.

Convergence of the remap method in L_x/L_y should be checked (particularly by for the modes with lowest k_ζ) by decreasing $\Delta k_\psi / k_{\zeta\mathrm{min}}$ (increasing N_x and p (IKXSPACE) whilst holding N_{mod} constant (defined later)).

4.3 Purely toroidal sheared rotation in general geometry

For this section only, we adopt the superscript ^{L} to represent quantities in the laboratory frame. All quantities without a superscript should be interpreted as being in the rotating frame (as in all other sections). The same coupling condition is derived by two routes to make clear the relationship between the frames.

The rotating frame is constructed [20] such that rigid body rotation Ω of the frame matches the plasma rotation ω_ϕ^L on the local flux surface. The angular rotation then transforms as

$$\omega_\phi(\psi) = \omega_\phi^L(\psi) - \Omega \tag{4.18}$$

such that $\omega_\phi(\psi) = 0$ (and hence $\mathbf{v_s} = 0$) at the centre of the radial domain.

4.3.1 In comoving frame

For a purely toroidal rotation, decomposing the toroidal flow into its parallel and perpendicular components gives

$$u_\parallel \mathbf{b} + \mathbf{v_s} = s_\mathrm{B} R \omega_\phi(\psi) R \nabla\varphi, \tag{4.19}$$

where φ is the toroidal angle. Since Eq. 4.19 is written in the comoving frame, it reads $0 + 0 = 0$ in the centre of the flux tube. From the normalisations above, one can show that in the normalised units

$$\mathbf{v_s} = \frac{1}{2}\rho_*^2 \frac{\mathbf{b}\times\nabla\psi}{B}\frac{\partial\bar{\Phi}}{\partial\psi}. \tag{4.20}$$

Taking the bi-normal component of Eq. 4.19 one finds

$$u_\parallel \underbrace{\mathbf{b}\cdot\nabla\zeta}_{=0} + \frac{1}{2}\rho_*^2 \underbrace{\frac{\mathbf{b}\times\nabla\psi}{B}\cdot\nabla\zeta}_{=2\mathcal{E}^{\psi\zeta}} \frac{\partial\bar{\Phi}}{\partial\psi} = s_\mathrm{B} R^2 \omega_\phi(\psi) \underbrace{\nabla\phi\cdot\nabla\zeta}_{=-1/2\pi R^2}, \tag{4.21}$$

hence

$$\frac{1}{2}\rho_*^2 \frac{\partial\bar{\Phi}}{\partial\psi} = -\frac{s_\mathrm{B}}{4\pi}\frac{1}{\mathcal{E}^{\psi\zeta}}\omega_\phi(\psi), \tag{4.22}$$

which upon differentiation gives the coupling relation

$$\underbrace{\frac{1}{2}\rho_*^2 \frac{\partial^2\bar{\Phi}}{\partial\psi^2}}_{\gamma_E} = -\frac{s_\mathrm{B}}{4\pi}\left[\frac{1}{\mathcal{E}^{\psi\zeta}}\underbrace{\frac{\partial\omega_\phi}{\partial\psi}}_{\mathrm{u}'} + \underbrace{\omega_\phi}_{=0}\frac{\partial}{\partial\psi}\left(\frac{1}{\mathcal{E}^{\psi\zeta}}\right)\right]. \tag{4.23}$$

Note that the second term on the right is zero when evaluated at the centre of the flux tube. All quantities in the above equation are flux functions, hence the shear rate will always be a flux function, as required for the discrete remapping method.

4.3.2 Relation to lab frame

In the laboratory frame Eq. 4.19 becomes

$$u_{\parallel}^{L}\boldsymbol{b}+\mathbf{v}_{s}^{L}=s_{\mathrm{B}}R\omega_{\phi}^{L}(\psi)R\nabla\varphi. \tag{4.24}$$

Taking the binormal component leads to

$$\frac{1}{2}\rho_{*}^{2}\frac{\partial\bar{\Phi}^{L}}{\partial\psi}=-\frac{s_{\mathrm{B}}}{4\pi}\frac{1}{\mathcal{E}^{\psi\zeta}}\omega_{\phi}^{L}(\psi), \tag{4.25}$$

and taking the radial derivative gives

$$\frac{1}{2}\rho_{*}^{2}\frac{\partial^{2}\bar{\Phi}^{L}}{\partial\psi^{2}}=-\frac{s_{\mathrm{B}}}{4\pi}\left[\frac{1}{\mathcal{E}^{\psi\zeta}}\frac{\partial\omega_{\phi}^{L}}{\partial\psi}+\omega_{\phi}^{L}\frac{\partial}{\partial\psi}\left(\frac{1}{\mathcal{E}^{\psi\zeta}}\right)\right], \tag{4.26}$$

which has the same form as Eq. 4.23 but different values.

To make explicit the relationships between the quantities, the previous equation is rewritten (using Eq. 4.18) in terms of the rotating frame quantities. Since the frame rotates rigidly, $\frac{\partial\Omega}{\partial\psi}=0$ and $\partial\omega/\partial\psi=\partial\omega^{L}/\partial\psi=u'$. The tensor component $\mathcal{E}^{\psi\zeta}$ is invariant under the transformation. The electric field transforms [20] as

$$\bar{\Phi}=\bar{\Phi}^{L}+s_{\mathrm{B}}s_{\mathrm{j}}\frac{2}{\rho_{*}^{2}}\Psi\Omega, \tag{4.27}$$

where Ψ is the (frame independent) poloidal flux with $\nabla\Psi$ radially outward, and the factor of $2/\rho_{*}^{2}$ arises from the normalisation of $\bar{\Phi}$ (see Sec. 2.2.6 for clarification of signs). The coupling condition can then be written as

$$\underbrace{\frac{1}{2}\rho_{*}^{2}\frac{\partial^{2}\bar{\Phi}}{\partial\psi^{2}}-s_{\mathrm{B}}s_{\mathrm{j}}\Omega\frac{\partial^{2}\Psi}{\partial\psi^{2}}}_{\gamma_{E}^{L}}=-\frac{s_{\mathrm{B}}}{4\pi}\left[\frac{1}{\mathcal{E}^{\psi\zeta}}\frac{\partial\omega_{\phi}}{\partial\psi}+\Omega\frac{\partial}{\partial\psi}\left(\frac{1}{\mathcal{E}^{\psi\zeta}}\right)\right], \tag{4.28}$$

when evaluated in the centre of the flux tube where $\omega_{\psi}=0$. Since $\Psi=f(\psi)$ only, from the relation 2.25 it follows that

$$\frac{\partial}{\partial\psi}\left(\frac{1}{\mathcal{E}^{\psi\zeta}}\right)=s_{\mathrm{j}}4\pi\frac{\partial^{2}\Psi}{\partial\psi^{2}}, \tag{4.29}$$

hence cancellation gives

$$\underbrace{\frac{1}{2}\rho_{*}^{2}\frac{\partial^{2}\bar{\Phi}}{\partial\psi^{2}}}_{=\gamma_{E}}=-\frac{s_{\mathrm{B}}}{4\pi}\frac{1}{\mathcal{E}^{\psi\zeta}}\underbrace{\frac{\partial\omega_{\phi}}{\partial\psi}}_{=u'}, \tag{4.30}$$

which is the same as Eq. 4.23 when evaluated at the centre of the flux tube. The shearing rates in the two frames are therefore related by

$$\gamma_{E}^{L}=\gamma_{E}-s_{\mathrm{B}}s_{\mathrm{j}}\Omega\underbrace{\frac{\partial^{2}\bar{\Psi}}{\partial\psi^{2}}}_{\mathcal{M}}, \tag{4.31}$$

but presently only γ_{E} is implemented in the code.

4.4 Poloidal asymmetries induced by anisotropic minorities

In this section, all units are dimensional, except those indicated with a superscript N, which denotes a dimensionless quantity using the GKW normalisations defined in Section 2.1.

Recent experiments have observed poloidal asymmetries in heavy impurities in the presence of ion cyclotron resonance heating (ICRH) of a light minority ion [78]. This phenomena is believed to be due to the anisotropy of the heated species, which sets up an equilibrium potential.

4.4.1 Reinke model

In the rotating frame of reference, the parallel force balance for an anisotropic species m (generalisation of Eq 3 of Ref. [78] with a sign correction for the second term) is

$$\nabla_{\parallel} p_{\parallel} - \frac{p_{\parallel} - p_{\perp}}{B} \nabla_{\parallel} B + n_m Z_m e \nabla_{\parallel} \Phi - n_m m_m \Omega^2 R \nabla R = 0 \tag{4.32}$$

Assuming $\nabla_{\parallel} T_{\parallel} = 0$, one finds on integration

$$n_m = A B^{-\eta} \exp\left(-\frac{e Z_m \Phi}{T_{\parallel}} + \frac{m_m \Omega^2 R^2}{2 T_{\parallel}} \right) \tag{4.33}$$

where $\eta = T_{\perp}/T_{\parallel} - 1$, and A is a constant of integration (with dimensions of $[nB^{\eta}]$) to be determined. In Ref. [78], $A = \{n\}/\{B^{-\eta}\}$, but this choice is actually inconsistent with the definition of the flux surface average $\{\}$; the correct definition (when $\Omega = 0$) would actually require

$$A_{\mathrm{FSA}} = \frac{\{n_m\}}{\{B^{-\eta} \exp(-e Z_m \Phi / T_{\parallel})\}}. \tag{4.34}$$

Although the difference between this and the normalisation of Eq. 3 of Ref. [78] is small for typical parameters, the deviation is enough to break quasi-neutrality and so the approximate normalisation of Ref. [78] is not possible to use in a code which contains a field solver. Furthermore, defining the constant A correctly as in Eq. 4.34 would require repeated iterations of the root finding quasi-neutrality solver condition over the whole flux surface, which although possible is probably unnecessary for this problem. Instead, to be consistent with the previous definition of n_{R0} as the density at $R = R_0$, we must have $A = n_{R0} B_{R0}^{\eta}$ and thus we have

$$n_m = n_{R0,m} \left(\frac{B}{B_{R0}} \right)^{-\eta} \exp\left(-\frac{e Z_m \Phi}{T_{\parallel}} + \frac{m_m \Omega^2 (R^2 - R_0^2)}{2 T_{\parallel}} \right) \tag{4.35}$$

For the minority species it is also convenient to define a generalised poloidal asymmetry energy

$$\mathcal{E}_{\Omega,\eta} = \mathcal{E}_{\Omega} + T_{\parallel} \eta \ln(B/B_{R_0}) \tag{4.36}$$

which in the GKW normalisations is

$$\mathcal{E}_{R,\eta}^N = \mathcal{E}_R^N + T_{\parallel}^N \eta \ln(B/B_{R_0}) \tag{4.37}$$

where $T_{\parallel}^N = T_{\parallel}/T_R^N T_{\mathrm{ref}}$ (set $= 1$ in the approximations below). This fits into the previous formalism such that

$$n_m = n_{R0} \exp(-\mathcal{E}_{\Omega,\eta}/T_{\parallel}) \tag{4.38}$$

The definition of $\mathcal{E}_{\Omega}$ is unchanged for all other species (although the value of Φ inside $\mathcal{E}_{\Omega}$ will be).

4.4.2 Bilato model

The Reinke model makes another approximation, by taking $p_{\perp}$ as a flux function. The self-consistent solution worked out by Bilato [79] gives a reduction in the asymmetry compared to the Reinke model roughly equivalent to transforming $\eta \to \eta^{3/4}$. This model is also implemented in GKW, and is selected by using an input value of $\frac{T_{\perp R0}}{T_{\parallel}} < 0$, where the absolute value is used by the code.

The exact solution is

$$n_m = n_{R0,m} \frac{T_{\perp}(\theta)}{T_{\perp R0}} \exp\left(-\frac{e Z_m \Phi}{T_{\parallel}} + \frac{m_m \Omega^2 R^2}{2 T_{\parallel}} \right) \tag{4.39}$$

where

$$\frac{T_\perp(\theta)}{T_{\perp R0}} = \left[\frac{T_{\perp R0}}{T_\parallel} + \left(1 - \frac{T_{\perp R0}}{T_\parallel}\right)\frac{B_{R0}}{B}\right]^{-1}. \tag{4.40}$$

The generalised asymmetry energy for the minority is then defined such that Eq. 4.38 still holds, with

$$\mathcal{E}_{\Omega,\eta} = \mathcal{E}_\Omega + T_\parallel \ln(T_{\perp R0}/T_\perp(\theta)) \tag{4.41}$$

which in the GKW normalisations is

$$\mathcal{E}^N_{R,\eta} = \mathcal{E}^N_R + T^N_\parallel \ln(T_{\perp R0}/T_\perp(\theta)) \tag{4.42}$$

where $T^N_\parallel = T_\parallel/T^N_R T_{\rm ref}$ (set $= 1$ in the approximations below).

4.4.3 Transformations of input parameters

If, as is usual in GKW, we take $R_0 = R_{\rm LFS}$, then the consequence is that the minority fraction $f_{\rm min} = \{n_m\}/\{n_e\}$ differs strongly from the value of $n_{R0,m}/n_{R0,e}$ in the presence of anisotropy.

However, in some cases, (particularly those of a strongly anistropic minority species) it is desirable that the code density inputs can be specified in the form of flux surface averages, because this is the canoncial conserved quantity in the absence of radial transport. Such an input specification also removes the need for post-processing of the output quantities as dsecribed in B.0.1.

For the reasons given in the preceding section, defining $n_{\rm input} = \{n\}$ and $rln_{\rm input} = R_{\rm ref}/L_{\{n\}} = 1/\{n\}\partial\{n\}/\partial\psi$ using the flux surface averaged densities requires an extra layer of iterative numerical solutions.

ITERATIVE transformation scheme under development, to be documented shortly....

4.4.4 Implementation

Using the generalised quasi-neutrality root finding solver for $Q(\Phi) = 0$ described in Sec. 4.1 above, it is then straightforward to insert the additional $B^{-\eta}$ dependence for a minority species. Since GKW also requires and computes radial derivatives of Φ, a maximum of 7 parameters are potentially required:

$$(n_{\rm input,m}/n_{\rm ref}, \quad T_\perp R0/T_\parallel, \quad \partial(T_{\perp R0}/T_\parallel)/\partial\psi, \quad T_\parallel, \quad \partial T_\parallel/\partial\psi, \quad Z_m, \quad m_m). \tag{4.43}$$

We remind the reader that $\psi = r/R_{\rm ref}$ is the dimensionless radial coordinate used in GKW. Note that the minority mass is not required as an input in the case when $\Omega = 0$.

In addition, it can be useful to include the asymmetric potential generated by the minority species without simulating it as a separate gyrokinetic species (combination method: NOT PRESENTLY WORKING CORRECTLY). For this purpose, although treated separately in the background quasi-neutrality solve, the minority species is after-wards combined with the bulk ion species (i). Since $Ze\Phi/T_\parallel \ll 1$ and $f_{\rm min} \ll 1$ only ever appear in combination if the exponential is expanded as series, the values of $T_\parallel$ and $\partial T_\parallel/\partial\psi$ make a negligible impact on the results (tested for $f_{\rm min} = 0.1$ and $T_\parallel/T_{\perp R0} = 10.0$). For simplicity, these parameters are therefore neglected in the combination method, and where required, the code makes the assumption that $T_i = T_\parallel$. (This assumption is required to have a value in the code, but in reality it does not affect the results). Only one mass is used for the combined species. Since the mass does not appear in the anisotropy effect, the average mass of the minority plus bulk should be given to model the centrifugal and kinetic effects correctly, but using the assumption $m_m = m_i$ will likely give almost identical results for most parameters. Outside the quasi-neutrality solve, the two species are combined, with $n_{R0,t} = n_{R0,i} + n_{R0,m}Z_m/Z_i$ such that $n_t = n_i + n_m Z_m/Z_i$ everywhere, which gives a resulting asymmetry energy of

$$\mathcal{E}_{\Omega,\eta,t} = -\ln\left(n_{R_0,i}\exp(-\mathcal{E}_{\Omega,i}) + \frac{n_{R_0,m}Z_m}{Z_i}\exp(-\mathcal{E}_{\Omega,\eta,m})\right) + \ln(n_{R_0,t}) \tag{4.44}$$

such that quasi-neutrality is still satisfied everywhere for the reduced set of species.

Because the combination option means that the asymmetry does not require an additional gyrokinetic species, the input parameters $n_{\mathrm{input,m}}/n_{\mathrm{ref}}, T_{\perp R0}/T_{\|}, \partial(T_{\perp R0}/T_{\|})/\partial\psi^N, Z_{\mathrm{min}}$ are separated from the individual species inputs and appear in the *SPCGENERAL* namelist as the array *icrh_params(1,2,3,4)*. respectively.

If instead the minority is treated as a full gyrokinetic species (independent method), the species is identified with *background='ICRH'* in the species namelist. This allows the fully general case without the (mild) assumptions on the parameters $m_m, T_{\|}$ and $\partial T_{\|}/\partial\psi^N$ required for the combination option. In this case, $n_{\mathrm{input,m}}/n_{\mathrm{ref}}$ and Z_{min} are taken from the species inputs *dens* and *Z* respectively (the first and last inputs *icrh_params(1,4)* are not used), and the species inputs *temp* and *rlt* describe $T_{\|}^N$. In the gyrokinetic equation, this species is then treated as a Maxwellian with temperature $T = T_{\|}$ (i.e., there is no bi-Maxwellian option implemented for F_0). Strictly, a more accurate approximation would be a Maxwellian with $T = (T_{\|}T_{\perp}^2)^{1/3}$, which might be more accurate for collisions or neoclassics.

After the solver completes, the code runs self-consistency checks on the quasi-neutrality of the gradients of the flux surface averages, which is a strong test on the consistency of the solver, using both the diagnostics in Eq. 4.6, and the transformations described in Sec. B.0.1. The gradient of the flux surface average

$$-\frac{1}{\{n\}}\frac{\partial\{n\}}{\partial\psi} = \left\{1/L_n^{e1}\right\} \tag{4.45}$$

for the minority is calcuated using

$$\frac{1}{L_n^{e,N}} = \frac{\partial\mathcal{E}_{R,\eta}}{\partial\psi} + \mathcal{E}_R\frac{1}{L_T^N} + \frac{1}{L_n^N}\bigg|_{R_0} \tag{4.46}$$

(note, the second term does NOT use the generalised energy $\mathcal{E}_{R,\eta}$) with

$$\frac{1}{L_n^{e1,N}} = \frac{1}{L_n^{e,N}}\frac{\exp(-\mathcal{E}_R)}{\left\{\exp(-\mathcal{E}_R)\right\}}, \tag{4.47}$$

which is the generalisation of Eq. 4.7). The first term in Eq. 4.46 is evaluated with the derivatives appearing in Sec. B.0.1, or Sec. ??, and the results of the checks are output to screen.

Examples of anisotropic minority input setup

Since the quasi-neutrality solver is only activated for centrifugal effects, the switch *cf_trap*=*T* must also be set even in the absence of rotation. Here we present two example input setups for a 5% helium-4 minority with $T_\perp/T_\parallel = 4$ and $\partial(T_\perp/T_\parallel)/\partial\psi = -8.5$. First treating the minority species as a gyrokinetic species, (independent method) we have:

```
&GRIDSIZE
number_of_species = 4
/
&SPCGENERAL
icrh_params = 999, 4.0, -8.5, 999  ! First and last not used
/
&SPECIES ! D ions
mass = 1.0000, dens = 0.90, z = 1.00, rlt = 5.5, rln = 2.5, temp = 1.0, uprim = 0.0
/
&SPECIES ! electrons
mass = 2.7e-4, dens = 1.00, z = -1.0, rlt = 4.5, rln = 2.5, temp = 1.2, uprim = 0.0
/
&SPECIES ! Molby trace impurity
mass = 48.000, dens = 0.00, z = 32.0, rlt = 6.5, rln = 1.5, temp = 1.0, uprim = 0.0
/
&SPECIES ! He ICRH minority
mass = 2.0000, dens = 0.05, z = 2.00, rlt = 9.9, rln = 2.5, temp = 2.0, uprim = 0.0
background = 'ICRH'
/
&ROTATION
cf_trap = .true.
/
```

or, alternatively, treating the minority in combination with the bulk ions (combination method), and giving practically identical results for Φ `and its deriviatives`:

```
&GRIDSIZE
number_of_species = 3
/
&SPCGENERAL
icrh_params = 0.05, 4.0, -8.5, 2.0  ! minority fraction, Tp/T||, d(Tperp/T||)/dpsi, Z_m
/
&SPECIES ! D ions + He minority (treated with Z = 1 in Possion and GK equations)
mass = 1.0527, dens = 1.00, z = 1.00, rlt = 5.5, rln = 2.5, temp = 1.0, uprim = 0.0
/
&SPECIES ! electrons
mass = 2.7e-4, dens = 1.00, z = -1.0, rlt = 4.5, rln = 2.5, temp = 1.2, uprim = 0.0
/
&SPECIES ! Molby trace impurity
mass = 48.000, dens = 0.00, z = 32.0, rlt = 6.5, rln = 1.5, temp = 1.0, uprim = 0.0
/
&ROTATION
cf_trap = .true.
/
```

Tests

- The combination method and the independent method give identical asymmetries.
- Quasi-neutrality is always satisfied everywhere, with all η, and all Ω, all geometries, for both R_0 choices, and arbitrary non-trace species combinations.
- Quasi-neutrality in the gradients is satisfied at all locations, and in the flux surface average density, checked using the formulae of Sec. B.0.1.
- The Φ asymmetry scales linearly with f_m, Z_m and η, and is practically independent of $T_\parallel$.
- High-Z impurity asymmetries are of the same magnitude as in Ref. [78] for the given parameters.

Chapter 5

Tearing modes

Here are some brief notes that describe how an imposed magnetic island is included in GKW (self-consistent tearing modes can occur naturally given correct inputs and thus do not require a description of their implementation here). The previous version (Refs [27, 28]) assumed that the island is always initialised in the largest poloidal mode. To have the island in a larger wavevector there are some subtleties to the initialisation. For a recent description of the implementation and benchmarks in slab geometry, see Zarzoso et al,. PoP, 2015.

5.1 Spectral implementation of magnetic islands in toroidal geometry

If we initially assume a constant poloidal flux perturbation with a given helicity

$$A_{\|} = A_{\|0} \exp[2\pi \mathrm{i}(ms - n\gamma)] \tag{5.1}$$

where s and γ are the poloidal and toroidal angles, respectively. Introducing the helical coordinate $\zeta = qs - \gamma$ and expanding the safety factor up to first order

$$q = m/n + \Delta\psi \partial q/\partial\psi \tag{5.2}$$

in the centre of the box (i.e. the mode is resonant at this position). We need to transform this representation to the (ψ, ζ, s) coordinate system which gives,

$$A_{\|} = A_{\|0} \exp[2\pi \mathrm{i}(n\zeta - \frac{\partial q}{\partial \psi} ns\Delta\psi)] \tag{5.3}$$

From which it directly follows that

$$k_\zeta = 2\pi n \rho_* \tag{5.4}$$

is the toroidal mode of the island. However for generality we assume that this may not be the largest wavelength within the domain, this is $k_\zeta^{min} = 2\pi\rho_*$.

In the helical field line co-ordinate system the x-points only appear in the ζ direction. In the toroidal and poloidal angles, x-points appear in both. From the periodicity constraint connected with the parallel boundary conditions it follows that the radial wavevectors are given by

$$k_\psi = (p/i_k) k_\zeta \frac{\partial q}{\partial \psi} = (p/i_k) 2\pi\rho_* \frac{\partial q}{\partial \psi} \tag{5.5}$$

where p is a set of integers in the range $[-N_x/2, N_x/2]$ and i_k is the parameter that determines the radial mode spacing. From $p = 1$ one directly obtains the radial size of the box

$$\psi = \left[-\frac{i_k}{2\partial q/\partial\psi}, \frac{i_k}{2\partial q/\partial\psi} \right] \cdot \frac{1}{k_\zeta^{min}} \tag{5.6}$$

The Fourier amplitudes of the different radial modes then can be calculated by integrating over the radial domain

$$\begin{aligned}\hat{A}(k_\psi, k_\zeta, s) &= A_{\|0} \exp\left(i\ nk_\zeta^{min}\zeta\right) \int \mathrm{d}\Delta\psi \exp\left[i\Delta\psi(k_\psi + nsk_\zeta^{min} dq/d\psi)\right] \\ &= \frac{A_{\|0} i_k}{\pi n \partial q/\partial\psi} \frac{\sin[\pi(ni_k s + p)]}{ni_k s + p} \\ &= C\frac{i_k}{n}\frac{\sin[\pi s^\circ]}{s^\circ} \end{aligned} \tag{5.7}$$

where we have used the ballooning angle $s^\circ = ni_k s + p$ in the last step and n is the poloidal wave mode (determined by the input parameter *isl_mode*).

One point to note at this point is that this perturbation is not periodic in the radial domain of GKW for all values of s. This manifests itself as a discontinuity in the parallel vector potential at the radial edge of the computational domain. A solution to this problem is to relax the constant flux approximation and therefore write the perturbation as,

$$A_{\|} = C\frac{i_k}{n} \exp ik_\zeta\zeta \sum_{p=N_x/2}^{N_x/2} A^p(s) \exp ipk_\psi\Delta\psi \tag{5.8}$$

Where $A^p(s)$ includes a damping function. We have chosen a Gaussian profile to reduce any ringing. This gives

$$A^p(s) = \exp\left((-(ni_k s + p)^2/L_s^2)\right)\frac{\sin\pi(ni_k s + p)}{(ni_k s + p)} \tag{5.9}$$

Where L is half width of the gaussian envelope. This approximation leads to a more satisfactory shape for the vector potential, where the value of L can be tuned to that the damping of the small k modes is negligible and the of high k modes sufficiently strongly to prevent the occurance of a jump at the boundary.

Thus the full implemented vector potential has the form,

$$A_{||} = C\frac{i_k}{n} \exp\left(ik_\zeta\zeta - i\omega t\right) \sum_{p=N_x/2}^{N_x/2} \exp\left(-\frac{(ni_k s + p)^2}{L_s^2}\right)\frac{\sin\pi((ni_k s + p))}{(ni_k s + p)} \exp\left(ipk_\psi\Delta\psi\right) \tag{5.10}$$

where ω is the poloidal rotation frequency of the island. Note that only a single binormal mode is used. The island half-width is defined in dimensional units (Wesson) as

$$W = 2\sqrt{\frac{rq\tilde{B}_r}{mB_p(dq/dr)}} = 2\sqrt{\frac{r\tilde{A}_\|}{B_p\hat{s}}} = 2\sqrt{\frac{qR\tilde{A}_\|}{B_t\hat{s}}} \tag{5.11}$$

where B_p is the poloidal component of the background magnetic field[1]. The island width determines the amplitude of the island perturbation: The factor $dq/dr = q\hat{s}/r$ is obtained from the magnetic shear and m is the poloidal mode number. Using the fact that the peturbed radial magnetic field is prescribed as $\tilde{B}_r = m\tilde{\Psi}/rR$, and that for the $s-\alpha$ model equilibrium the perturbed magnetic flux is given by $\tilde{\Psi} = R\tilde{A}_{\|0}$, and $q = rB_t/RB_p$, the amplitude of the island vector potential (now in GKW normalised units) is given by

$$\tilde{A}_{\|0N} = \frac{W_N^2 B_{tN}\hat{s}}{4qR_N}, \qquad C = \frac{W_N^2 B_{tN}\epsilon}{4\pi q^2 R_N}. \tag{5.12}$$

where B_t^N is taken at the low field side in order to define the island width at the same location, and $W = W_N\rho_{\rm ref}$. Note that the amplitude of C given in equation 5.12 is divided by 2, due to the form of the Fourier representation described in Sec. 2.3.

[1]The correspondance with the literature on the sheared slab by Smolyakov and others is as follows: They have $W^2 = 4L_s A_\|/B_0$, but ψ is the vector potential in their notation. The toroidal result used in GKW can be obtained by inserting $1/L_s = (1/q)(dq/dr)$, and $B_0 = B_p$.

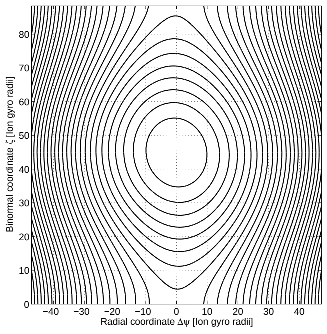


Figure 5.1: Magnetic Island Geometry for an island with a half width of 30 ion gyro-radii

To run GKW with a magnetic island present the following options must be used: Firstly, the code must be set to run electro-magnetically by setting *nlapar=.true.*. In the *SPCGENERAL* namelist, *tearingmode* must be set to *.true.*, which adds the magnetic island as a perturbation of the equilibrium field. The half width of the island is determined by the input parameter *wstar* which is the island width normalised to ρ_{ref} (usually the ion gyro-radius). A point of note is that this island width can not exceed (and should be smaller than) the radial extent of the computational box due to the periodicity constraints of a pseudo-spectral code. The rotation frequency of the island (ω) is set by the parameter *isl_rot_freq*, normalised as other frequencies in the code BUT with opposite sign (positive is in the $\nabla\zeta$ direction, which is the electron grad-B drift direction when $s_j = 1$ and rln ¿ 0. By default the island is placed in the smallest non-zero poloidal wavevector (*isl_mode=2*), however this can be changed by setting *isl_mode* to another integer (e.g. *isl_mode=3* will give 2 islands within the domain as the zero mode is always the first mode).

We recommend initialising the distribution function with *finit='zero'* as the island itself will seed the turbulence, and this gives faster convergence compared to non-zero distribution function initialisations. When plotting XY profile outputs, setting *lisl_follow* in the *DIAGNOSTIC* namelist will follow the magnetic island so that the O-point is always in the centre of the box. We do not recommend using the options to initialise the island after the turbulence, as growing the island too quickly drives a unphysical and large zonal flow, which initally supresses the turbulence and then takes a long time to decay.

```
&SPCGENERAL
beta = 0.0001,  ! not used
tearingmode=.true.,  wstar = 5.0,  finit = 'zero',  adiabatic_electrons = .false.
Ls = 2.0,  isl_mode = 2, isl_rot_freq = 0.1
/
```

5.2 Spectral implementation of islands in 2D sheared slab geometry

An alternative way to deal with 2D islands is also implemented, in which parallel dynamics are projected into the y direction, as is usually done in slab analytics. This allows GKW islands to be compared directly with analytical results in the slab. Whilst this setup is intended for use with the slab geometry, it might also be applicable for a LFS toroidal case. The island amplitude is decoupled from geometric magnetic shear, and a new parameter, *isl_shear* is used in determing the island width. To use this option the geometric shear *shat* is set to 0 (which means reducing the coordinate system to purely Cartesian) and the code is run with a single s point and without parallel dynamics (a limit in which the geometrical magnetic shear plays no role). This also implies that the minimum k_x and k_y now can be set completely independently, but they are chosen to be the same (to give a square box).

The perturbed $A_\parallel$ in this case then also contains the part of the island that normally comes from the background magnetic field (in the 2D case the background B plays no role in the geometry and theoretically only enters via the Larmor radius). The slab geometry is defined in this case through a Cartesian set of coordinates, x,y and z, where z, the direction of the guide magnetic field, is meant to be a symmetry coordinate for the whole system, i.e. $\partial/\partial z = 0$, accordingly to the existing dedicated literature. The total magnetic field in this case reads

$$\mathbf{B} = B_z \nabla z - \nabla\psi \times \nabla z, \tag{5.13}$$

with

$$\psi = -\frac{i_s x^2}{2} B_z + \tilde{\psi}\cos(k_y y), \tag{5.14}$$

where B_z is a constant and where i_s corresponds to *isl_shear*, the (inverse of the) scale of variation of B_y, which plays a role analogous to the magnetic shear in toroidal geometry. The y-dependent term represents the island. The amplitude $\tilde{\psi}$, assumed to be constant as a consequence of the well-known "constant-ψ approximation", is linked to the island width w through the relation

$$w^2 = \frac{4\tilde{\psi}}{i_s B_z}. \tag{5.15}$$

In the 3D case, only the latter "island"-part of ψ is imposed as a perturbation, entering the gyrokinetic equation through v_χ in the nonlinear term. In this 2D case, in constrast, we consider a shearless slab, with equilibrium

$$\mathbf{B} = B_z \nabla z \tag{5.16}$$

(constant) and we include the whole ψ of Eq. 5.14 as a perturbation. In Fourier space this extra island component (additional to the usual part in the island k_y mode) is applied in the $k_y = 0$ mode, and has the form

$$A_\parallel^+(p) = \frac{4(-1)^p i_s}{\bar{k}_x(p)^2}\exp(-(p/Ls)^2) \tag{5.17}$$

where p is again the radial mode number, and an exponential damping has again be added to the exact Fourier transformed quantity. The overbar on $\bar{k}_x$ represents the modes in the code, to distinguish them from the island wavenumbers k_x and k_y used above. While it is apparent that ψ is periodic on y, the quadratic dependence on x cannot be exactly reproduced in Fourier space, as its derivative is not the same on the two boundaries. Thus, analogously to the toroidal case, we again employ the exponential damping for the high k_x modes in order to remove boundary discontinuities. This approximation has the effect of creating a second magnetic island at the boundary which in full 3D geometry is obscured by parallel coupling, ballooning and curvature. The effect of this boundary island can be minimised by choosing an appropriate combination of the parameters box size, shear, island width, damping length (which differ in slab and torodial geometry) but it can never be completely eliminated.

In the 2D case, there is however an additional island option, to add *two* identical islands radially with perfect periodicity. This option is activated when *shat* $= 0$ and *isl_shear* > 0 by selecting a negative *wstar*. By choosing this option, ψ is approximated by

$$\psi = \hat{C}\cos k_x x + \tilde{\psi}\cos k_y y, \tag{5.18}$$

where $\hat{C} = \pi B_z i_s / k_x^2$ is a constant chosen in order to have the same amplitude as the other case for $B_y = \partial\psi/\partial$ at the edge of the box. In this case, the additional nonzero component only appears in the $k_y = 0$, $p = -1, 1$ modes with value $A_\parallel^+ = \pi i_s / k_x^2$. The sinusoidal dependence can of course be implemented without any approximation, or artificial damping, leading therefore to no parasitic boundary island formation. However, the presence of both a maximum and a minimum in x corresponds to the presence of two identical islands with $\pi/2$-shift in the y-direction. By carefully choosing the box width, one is able to reduce the mutual interaction of the two islands.

Summarizing, for slab geometry GKW has three options for islands:

- To run with shat > 0 in 3D, which is expensive (and where discontinuities at the boundary create numerical problems) and which does not reproduce the boundary condition $\partial/\partial z = 0$ used in many analytic models.
- To run with shat $= 0$ and wstar > 0. These are 2D runs, much faster, but which still employ the artificial damping on high- $\bar{k}_x$ modes, and therefore require specific attention to the discontinuity at the boundary, as in the full 3D cases.
- To run with shat $= 0$ and wstar < 0. These are 2D runs, with no artificial damping, but which contain two identical out of phase magnetic islands at two radial locations.

Example 2D setup:

```
&GRIDSIZE
NX = 167,  N_mu_grid = 8,  n_vpar_grid = 12, number_of_species = 2,
N_s_grid = 1,  NMOD = 20,  nperiod = 1,
/
&MODE
mode_box = .true., krhomax = 0.45, ikxspace = 1,
/
&GEOM
GEOM_TYPE = 'slab_periodic', SHAT = 0.0,  Q = 1.0, EPS = 1.0,
/
&SPCGENERAL
finit = 'zero',  tearingmode=.true.,
wstar = 10.0, isl_rot_freq = -0.02,
adiabatic_electrons = .false.
ls=8.0, isl_shear = 0.1
```

5.3 Nonspectral implementation of a magnetic island

If the radial direction is not treated in Fourier space, the implementation is much more direct, since we need only to introduce at $t = 0$ the term

$$A_{\|} = A_{\|0} \cos\left(2\pi \mathrm{i}(ms - n\gamma)\right) \tag{5.19}$$

which can be expressed by

$$A_{\|} = \frac{1}{2} A_{\|0} \left[e^{i2\pi n\zeta} \left(\cos \frac{\partial q}{\partial \psi} ns\Delta\psi - i \sin \frac{\partial q}{\partial \psi} ns\Delta\psi \right) + e^{-i2\pi n\zeta} \left(\cos \frac{\partial q}{\partial \psi} ns\Delta\psi + i \sin \frac{\partial q}{\partial \psi} ns\Delta\psi \right) \right] \tag{5.20}$$

In the binormal direction we initialise only one mode, namely $e^{i2\pi n\zeta}$ and assume the existence of the other mode in other to get a real quantity once the inverse Fourier transform is performed. Therefore, we need to implement only the potential

$$A_{\|,+} = \frac{1}{2} A_{\|0} \left(\cos \frac{\partial q}{\partial \psi} ns\Delta\psi - i \sin \frac{\partial q}{\partial \psi} ns\Delta\psi \right) \tag{5.21}$$

In order to decouple the island from the boundary of the simulation box, we multiply the previous potential by a radial envelope having the explicit expression

$$A_r\left(\Delta\psi\right) = \frac{1}{4} \left(1 + \tanh\left(\frac{\Delta\psi + \Delta\psi_0}{\delta\psi_0} \right) \right) \left(1 - \tanh\left(\frac{\Delta\psi - \Delta\psi_0}{\delta\psi_0} \right) \right) \tag{5.22}$$

where $\Delta\psi_0$ and $\delta\psi_0$ are defined in the tearingmode namelist of the input file of GKW by the variables `psi_0` and `delta_psi_0`, respectively.

In a 2D sheared slab geometry, the previous implementation reduces to

$$A_\parallel = \frac{1}{2} A_{\parallel 0} A_r \left(\Delta\psi\right) \tag{5.23}$$

and the background magnetic shear is introduced together with this perturbation by means of the mode $\kappa_\zeta = 0$. Therefore, the total parallel vector potential reads in 2D sheared slab geometry

$$A_\parallel = \frac{B_z i_s}{2} \left[\Delta\psi^2 - \left(L_x/2\right)^2\right] + \frac{1}{2} A_{\parallel 0} A_r \left(\Delta\psi\right) e^{i2\pi n\zeta} \tag{5.24}$$

where the additional term $\frac{B_z i_s}{2} \left(L_x/2\right)^2$ is introduced to impose a vanishing potential at the boundary in the nonspectral version. We therefore need to introduce the term $\frac{B_z i_s}{2} \left[\Delta\psi^2 - \left(L_x/2\right)^2\right]$ in the $\kappa_\zeta = 0$ component and the term $\frac{1}{2} A_{\parallel 0} A_r \left(\Delta\psi\right)$ in the $\kappa_\zeta = \kappa_{\rm isl}$ component.

5.4 Reconstruction of flux tube radial profiles and profile flattening in GKW

The benchmark of the magnetic islands implementation includes the flattening of the density and pressure profiles. The equilbrium profiles are linear with respect to the radial coordinate and are present in the source terms. Since we are solving the gyrokinetic equation for the perturbed part of the distribution function, the solution should respond to the source terms due to the background. In particular, the magnetic island should give a perturbation that cancels the background out inside the separatrix. Note that runs must be non-linear for the correct level of flattening to be obtained. To reconstruct the radial profiles the XY diagnostics must enabled in the diagnostics namelist.

5.4.1 Density profiles

The files named `den0#_000*_*` hold the normalised XY perturbed density data ($\tilde{n}_{N,s}$) for each large time step. The background density profile can be reconstructed by using the local radial cordinate $x_r = \psi/\rho_* = r/\rho_{\rm ref}$ which is written in the file named *xphi* (with the same dimensions as the XY slices) and the input parameter *RLN_s* $= -(1/n_{R_0,s})\frac{\partial n_{R_0,s}}{\partial\psi}$. The total density of species s is $n_s = n_{s,\rm eq} + \tilde{n}_s$, where $n_{s,\rm eq} = n_{R_0,s}\left(1 - \frac{r-r_0}{L_{n,s}}\right)$ is the global density of the Maxwellian equilibrium written in the local limit, i.e. $n_{R_0,s}$ at the resonant position r_0 plus the linear variation due to the gradient at that position. Note that in the expression of $n_{s,\rm eq}$, the quantity $n_{R_0,s}$ does not exhibit any radial dependence since it is considered in the local limit. The perturbed density of species s is

$$\tilde{n}_s = \int_{\mathbb{R}^3} d^3\mathbf{v}\delta f_s = \rho_* n_{R_0,s} \int_{\mathbb{R}^3} d^3\mathbf{v}_{N,s}\delta f_{N,s} = \rho_* n_{R_0,s}\tilde{n}_{N,s} \tag{5.25}$$

Therefore, the total density for species s reads

$$n_s = n_{R_0,s}\left(1 - \frac{r-r_0}{L_{n,s}}\right) + \rho_* n_{R_0,s}\tilde{n}_{N,s} = n_{R_0,s}\left(1 - \frac{\left(x_r - x_{r0}\right)\rho_{\rm ref}}{L_{n,s}}\right) + \rho_* n_{R_0,s}\tilde{n}_{N,s} \tag{5.26}$$

We can now normalise the density for species s as $n_{N,s} = n_s/n_{R_0,s}$

$$n_{N,s} = \left(1 - \frac{\left(x_r - x_{r0}\right)\rho_{\rm ref}}{L_{n,s}}\right) + \rho_* \tilde{n}_{N,s} \tag{5.27}$$

and define the change in the density for species s as $\Delta n_{N,s} = n_{N,s} - 1$

$$\Delta n_{N,s} = -\left(x_r - x_{r0}\right) \frac{\rho_{\text{ref}} R_{\text{ref}}}{L_{n,s} R_{\text{ref}}} + \rho_* \tilde{n}_{N,s} = -\rho_* \left(x_r - x_{r0}\right) \frac{R_{\text{ref}}}{L_{n,s}} + \rho_* \tilde{n}_{N,s} \tag{5.28}$$

or alternatively

$$\frac{\Delta n_{N,s}}{\rho_*} = \tilde{n}_{N,s} - \frac{R_{\text{ref}}}{L_{n,s}} \left(x_r - x_{r0}\right) \tag{5.29}$$

and the total density gradient (background plus perturbed density) can be computed as

$$\frac{R_{\text{ref}}}{L_{n_{\text{eq}}+\tilde{n}}} = -\frac{1}{n_{R_0,s}} \frac{\partial(n_{s,\text{eq}} + \tilde{n}_s)}{\partial \psi} = -\frac{\partial \tilde{n}_{N,s}}{\partial x_r} + \frac{R_{\text{ref}}}{L_{n,s}}. \tag{5.30}$$

5.4.2 Pressure profiles

To obtain the pressure profiles we need to use the files `ene0#_000*_*`, which hold the normalised second moments of the distribution function, namely

$$\tilde{E}_{N,s} = \int_{\mathbb{R}^3} d^3\mathbf{v}_{N,s} \delta f_{N,s} v_{N,s}^2 \tag{5.31}$$

It can be shown that the pressure perturbation is written in terms of this quantity as follows

$$\tilde{p}_s = \frac{1}{3} m_{\text{ref}} v_{th,s}^2 \rho_* n_{R_0,s} m_{R,s} \tilde{E}_{N,s} \tag{5.32}$$

Taking into account the definition of the reference temperature T_{ref}, one can write

$$\tilde{p}_s = \frac{2}{3} \rho_* n_{R_0,s} T_{\text{ref}} m_{R,s} v_{R,s}^2 \tilde{E}_{N,s} \tag{5.33}$$

In addition, $m_{R,s} v_{R,s}^2 = T_{R,s}$ and $T_s = T_{\text{ref}} T_{R,s}$, which leads to

$$\tilde{p}_s = \frac{2}{3} \rho_* n_{R_0,s} T_s \tilde{E}_{N,s} \tag{5.34}$$

In the same way as has been done with the density, the total pressure is defined as $p_s = p_{s,\text{eq}} + \tilde{p}_s$, where $p_{s,\text{eq}} = n_{R_0,s} T_s \left(1 - \frac{r-r_0}{L_{n,s}}\right) \left(1 - \frac{r-r_0}{L_{T,s}}\right)$. In the local limit we can neglect terms of second order in ρ_* and therefore, the equilibrium pressure profile reads

$$p_{s,\text{eq}} \approx n_{R_0,s} T_s \left(1 - \rho_* \left(x_r - x_{r0}\right) \frac{R_{\text{ref}}}{L_{n,s}} - \rho_* \left(x_r - x_{r0}\right) \frac{R_{\text{ref}}}{L_{T,s}}\right) \tag{5.35}$$

We can now divide by $n_{R_0,s} T_s$ and define the change in the normalized pressure as

$$\Delta p_{N,s} = \frac{2}{3} \rho_* \tilde{E}_{N,s} - \rho_* \left(x_r - x_{r0}\right) \frac{R_{\text{ref}}}{L_{n,s}} - \rho_* \left(x_r - x_{r0}\right) \frac{R_{\text{ref}}}{L_{T,s}} \tag{5.36}$$

or equivalently

$$\frac{\Delta p_{N,s}}{\rho_*} = \frac{2}{3} \tilde{E}_{N,s} - \left(x_r - x_{r0}\right) \frac{R_{\text{ref}}}{L_{n,s}} - \left(x_r - x_{r0}\right) \frac{R_{\text{ref}}}{L_{T,s}} \tag{5.37}$$

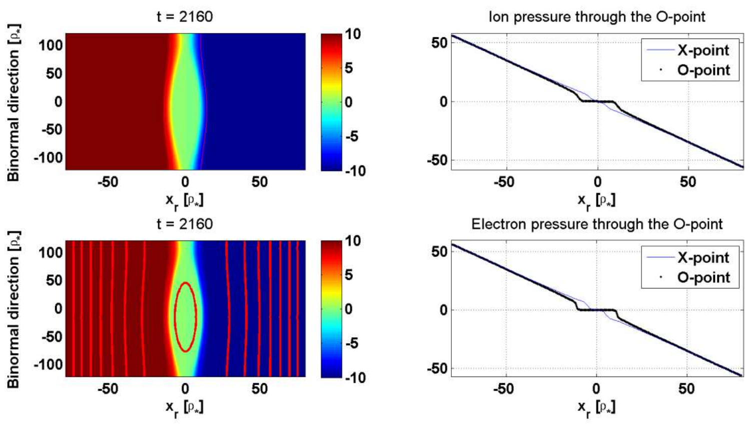


Figure 5.2: Left: 2D colormap of the pressure profiles for ions and electrons in the presence of a magnetic island. The isocontours of the vector potential are shown only with the electron colormap. Right: radial profiles of the pressure through the O (black line) and X (blue line) points.

This quantity is plotted in figure 5.2. Note that the flattening is observed in the moments of the distribution function, namely the density and the pressure. One can define a temperature for the unperturbed profiles, since these unperturbed quantities are obtained from a Maxwellian equilibrium. However, one cannot say that the total pressure is $p = nT$, where $n = n_{\rm eq} + \tilde{n}$ and $T_{\rm eq} + \tilde{T}$. For this reason, the flattening of the profiles can only be analysed for the density and the pressure. An example of unexpected results that one can obtain when defining a perturbed temperature is evidenced when introducing a magnetic island in a plasma with flat temperature profile but finite density gradient. One expects that the temperature is not modified, since it is already flat. However, by defining the temperature as $T = p/n$ one obtains a modification which actually corresponds to the modifications of both density and pressure.

The total pressure gradient (background plus perturbed pressure) reads

$$\frac{R_{\rm ref}}{L_{p_{\rm eq}+\tilde{p}}} = -\frac{1}{n_{R_0,s}T_s}\frac{\partial(p_{s,\rm eq}+\tilde{p}_s)}{\partial\psi} = -\frac{\partial}{\partial x_r}\frac{\Delta p_{N,s}}{\rho_*} = -\frac{2}{3}\frac{\partial \tilde{E}_{N,s}}{\partial x_r} + \frac{R_{\rm ref}}{L_{n,s}} + \frac{R_{\rm ref}}{L_{T,s}} \tag{5.38}$$

5.5 Frozen electrons

Due to the small inertia of electrons, they are considered as frozen particles to the magnetic field lines. This means in particular that the rotation frequency of electrons must equal the rotation frequency of the island, namely

$$\omega_{\rm isl} \equiv \omega_{\rm TOT,e} = \omega_{*,e} + \omega_{E\times B} \tag{5.39}$$

which constitutes the second benchmark after implementation of a magnetic island.

Therefore, the electric field must always compensate for the difference between the diamagnetic drift produced by any pressure gradient inside the island and the rotation of the island itself. In particular, if the island does not rotate, for a sufficiently large island resulting in a complete flattening, the electrostatic potential must vanish inside the island. To verify the previous identity, one must conveniently quantify from the GKW outputs the diamagnetic frequency resulting from the total pressure profile and the $E \times B$ frequency. These two quantities are derived hereafter assuming a slab geometry and considering that the binormal (in the y direction) structure is determined by the island. The diamagnetic frequency in the y direction reads

$$\omega_{*,s} = k_y \frac{\mathbf{b}\times\nabla p_s}{e_s n_s B}\cdot\mathbf{e}_y \approx k_y \frac{1}{e_s n_{s,\rm eq} B}\frac{\partial p_s}{\partial r} \tag{5.40}$$

Using the normalizations of GKW, we can write

$$\omega_{*,s} \approx k_y \frac{1}{e_{N,s} e n_{R_0,s} B_N B_{\rm ref}} \frac{n_{R_0,s} T_s \rho_*}{\rho_{\rm ref}} \frac{\partial}{\partial x_r} \frac{\Delta p_{N,s}}{\rho_*} \tag{5.41}$$

The temperature T_s can now be expressed in terms of $T_{\rm ref}$ and $T_{R,s}$ and the reference Larmor radius used to write

$$\omega_{*,s} \approx k_y \frac{1}{2} v_{\rm th,ref} \frac{T_{R,s}}{e_{N,s} B_N} \rho_* \left(\frac{2}{3} \frac{\partial \tilde{E}_{N,s}}{\partial x_r} - \frac{R_{\rm ref}}{L_{n,s}} - \frac{R_{\rm ref}}{L_{T,s}} \right) \tag{5.42}$$

Under the assumption that the y structure is determined by the island, one can write the wave number is that direction as $k_y = \pi / y_{\rm max}$, where $y_{\rm max}$ is the maximum value of the y coordinate. Expressing ρ_* as $\rho_* = \rho_{\rm ref}/R_{\rm ref}$ and normalizing the diamagnetic frequency to the transit frequency $\omega_t = v_{\rm th,ref}/R_{\rm ref}$ one gets the normalized diamagnetic frequency in slab geometry

$$\omega_{*,N,s} \approx \frac{\pi}{y_{\rm max,N}} \frac{1}{2} \frac{T_{R,s}}{e_{N,s} B_N} \left(\frac{2}{3} \frac{\partial \tilde{E}_{N,s}}{\partial x_r} - \frac{R_{\rm ref}}{L_{n,s}} - \frac{R_{\rm ref}}{L_{T,s}} \right) \tag{5.43}$$

In a similar way, one can show that the $E \times B$ frequency in GKW units reads

$$\omega_{E,N,s} \approx \frac{\pi}{y_{\rm max,N}} \frac{1}{2} \frac{1}{B_N} \frac{\partial \phi_N}{\partial x_r} \tag{5.44}$$

Figure 5.3 represents the profiles of the frequency for each species in the presence of an island rotating in the ion diamagnetic direction with a frequency $\omega_{\rm isl} = -0.006$. This frequency is represented by a solid black line, which must overlap the solid green line, corresponding to the total frequency of electrons. We observe this overlap everywhere inside the separatrix, but not around the X-point.

5.6 Dependence of the electrostatic potential on the island rotation frequency

The third and last benchmark needed after the implementation of a magnetic island constitutes the dependence of the electrostatic potential on the island rotation frequency. This dependence is calculated assuming that electrons inside the island short out any variation of the parallel electric field, i.e. we need to verify that $\nabla_\parallel \phi = -c^{-1} \partial_t \psi$ inside the island. For this we need to write an explicit expression for the parallel gradient, which is $\nabla_\parallel = \mathbf{b} \cdot \nabla$, where $\mathbf{b} = B^{-1}\mathbf{B}$ is the unit vector in the direction of the magnetic field. Let us perform the calculation, for the sake of simplicity, in sheared 2D slab geometry, where the magnetic field reads

$$\mathbf{B} = B_z \nabla z - \nabla \psi \times \nabla z = B_z \nabla z + \frac{B_z x}{L_s} \nabla y - \tilde{\psi} k_y \sin\left(k_y y - \omega_{\rm isl} t\right) \nabla x \tag{5.45}$$

Under the assumption that the amplitude of the magnetic field is not significantly modified by the presence of the magnetic island and considering no derivatives in the z-direction (slab model), we can write

$$\nabla_\parallel = -\frac{\tilde{\psi} k_y}{B_z} \sin\left(k_y y - \omega_{\rm isl} t\right) \partial_x + \frac{x}{L_s} \partial_y \tag{5.46}$$

The equation for the electrostatic potential is written as

$$\frac{\tilde{\psi} k_y}{B_z} \sin\left(k_y y - \omega_{\rm isl} t\right) \frac{\partial \phi}{\partial x} - \frac{x}{L_s} \frac{\partial \phi}{\partial y} = \frac{1}{c} \frac{\partial \psi}{\partial t} \tag{5.47}$$

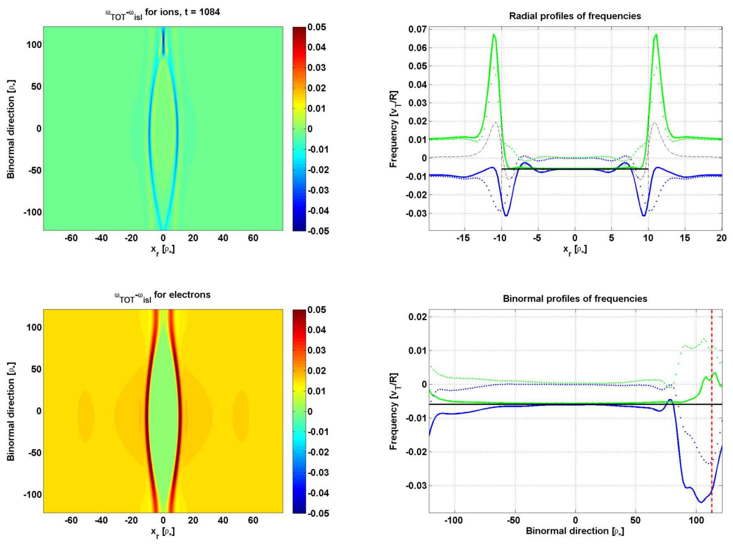


Figure 5.3: Left: 2D colormap of the frequency profiles for ions (up) and electrons (down) in the presence of a magnetic island with $R/L_n = 0.7$ and $R/L_T = 0$. Right: radial profiles of the frequency through the O-point (up) and binormal profiles of the frequency through the O and X points (down). Electrons are represented by green curves and ions by blue curves. The dotted lines represent the diamagnetic frequency and the solid lines the total frequency. For the radial profiles, the black dashed line represents the $E \times B$ frequency and the solid black line the rotation frequency of the island. For the binormal profiles, the dashed red line indicates the position of the X-point and the solid black line the rotation frequency of the island.

Noticing that x and y coordinates must be related to each other on a flux surface for a given time t by the flux label

$$\Omega = \frac{x^2 B_z}{2\tilde{\psi} L_s} - \cos\left(k_y y - \omega_{\mathrm{isl}} t\right) \tag{5.48}$$

we can rewrite the equation satisfied by the electrostatic potential as follows

$$-\frac{x}{L_s}\frac{\partial \phi}{\partial y} = \frac{1}{c}\frac{\partial \psi}{\partial t}, \quad \text{for } d\Omega = 0 \tag{5.49}$$

or equivalently

$$-\frac{x}{L_s}\left.\frac{\partial \phi}{\partial y}\right|_{\Omega} = \frac{1}{c}\frac{\partial \psi}{\partial t} \tag{5.50}$$

The derivative of x with respect to y, for $x \neq 0$, reads

$$\frac{dx}{dy} = -\frac{\tilde{\psi} L_s}{B_z x}\sin\left(k_y y - \omega_{\mathrm{isl}} t\right) \tag{5.51}$$

and therefore, the equation satisfied by the electrostatic potential becomes, for $k_y y \not\equiv_\pi \omega_{\mathrm{isl}} t$

$$\frac{k_y}{B_z}\left.\frac{\partial \phi}{\partial x}\right|_{\Omega} = \frac{\omega_{\mathrm{isl}}}{c} \tag{5.52}$$

which gives, after integration

$$\phi = \frac{\omega_{\mathrm{isl}} B_z}{k_y c}\left(x - h\left(\Omega\right)\right) \tag{5.53}$$

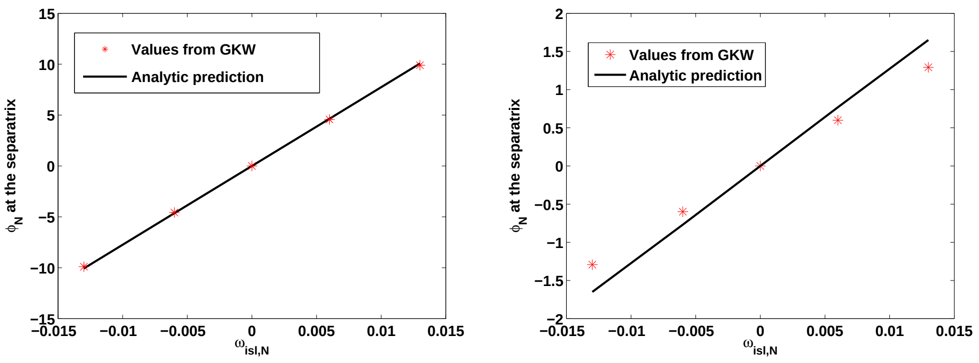


Figure 5.4: Expected values of the electrostatic potential at the separatrix (solid black line) and values obtained from simulations with GKW (red asterisks). Left: $w = 10$. Right: $w = 2$

where $h(\Omega)$ is a constant of integration for each flux surface. Note that the expression we have obtained for the electrostatic potential is only valid for $x \neq 0$ and $k_y y \not\equiv_\pi \omega_{\rm isl} t$. The function $h(\Omega)$ usually vanishes inside the island, corresponding to a complete flattening of radial profiles. Therefore, inside the separatrix and for $x \neq 0$ we can write, in GKW units

$$\phi_N = 2\frac{\omega_{\rm isl,N} y_{\rm max,N}}{\pi}\Delta x_r \tag{5.54}$$

where Δx_r is the distance to the resonant surface. The extension to $x = 0$ and $k_y y \equiv_\pi \omega_{\rm isl} t$ can be done by assuming that the electrostatic potential is continuous. Therefore, $\phi(x = 0) = \lim_{x\to 0^\pm}\phi = 0$, $\phi(x = w, k_y y \equiv_\pi \omega_{\rm isl} t) = \lim_{x\to w^+}\phi = 2\frac{\omega_{\rm isl,N} y_{\rm max,N}}{\pi} w$ and $\phi(x = -w, k_y y \equiv_\pi \omega_{\rm isl} t) = \lim_{x\to -w^-}\phi = -2\frac{\omega_{\rm isl,N} y_{\rm max,N}}{\pi} w$. The left panel of figure 5.6 shows a remarkable agreement between the expected values for the electrostatic potential at the separatrix ($\Delta x_r = w$) from the previous expression (solid black line) and the values obtained from five simulations with GKW (red asterisks). These simulations have been performed with $w = 10$, $y_{\rm max,N} = 121.565$ and flat profiles.

When the island width is comparable to or smaller than the ion Larmor radius, the ion pressure profile is little modified, whereas the electron pressure profile inside the island can still be modified. Of course, in the absence of background density and temperature profiles, this means that the ion pressure remains almost flat, whereas the electron pressure is only modified around the separatrix. However, this modification is of the order of the ion Larmor radius, meaning that for an island width of the order of the ion Larmor radius the electron pressure profile will be modified inside the island. In this case, the equation satisfied by the electrostatic potential should include the term related to the parallel gradient of the electrons pressure inside the island, namely

$$\nabla_\parallel \phi = -c^{-1}\partial_t \psi + \frac{1}{en_e}\nabla_\parallel p_e \tag{5.55}$$

Integrating this equation leads straightforwardly to the expression

$$\phi = \frac{\omega_{\rm isl} B_z}{k_y c}\left(x - h(\Omega)\right) + \frac{1}{en_e} p_e \tag{5.56}$$

or equivalently, in GKW units

$$\phi_{N,\rm sep} = 2\frac{\omega_{\rm isl,N} y_{\rm max,N}}{\pi} w + \left.\frac{\Delta p_{N,s}}{\rho_*}\right|_{\rm sep} \tag{5.57}$$

The right panel of figure 5.6 shows quite a good agreement, but some differences are observed. More analysis is needed for a complete understanding in the case of small islands.

Chapter 6

Eigenvalue solver

Here we give some information on the usage of the eigenvalue solver, implemented in GKW, as well as some technical details. Further information can be found in the presentation of R. Buchholz on the GKW webpages under Talks.

As the system size is to big for a direct (numerical) solution, projection methods are used. For a brief introduction, see the SLEPC manual [57] and references therein.

6.1 Usage

First, GKW must be compiled with SLEPC/PETSC, see 9.2.6 for details.

The easiest way to create an input file for the eigenvalue solver, is the following: First, set up an input file for use with the exponential time integration scheme. As second step then replace *METHOD = 'EXP'* with *METHOD='EIV'* in the *control* namelist, and set *METH=1* and *NAVERAGE=1* (also *METH=2* and/or *NAVERAGE>1* work, but these are at least not faster and usually slower).

The third and final step is to set the parameters for the eigenvalue solver itself. This is done using the optional namelist *eiv_integration*, an example of which is given below. One has to select *which_eigenvalues* should be searched for, for a full list of the options, see the sample input file. In addition, you can choose between two methods for extracting eigenvalues, `type_extraction = 'harmonic'`, and `type_extraction = 'ritz'`. The target values for `growthrate` and `freq`) are always used if `type_extraction = 'harmonic'`. They are also used if you choose `type_extraction = 'ritz'` and `which_eigenvalues = 11` and `comparison_routine = 2` ('TARGET_COMPLEX'). Becuase the harmonic extraction makes a kind of shift-and-invert transformation, values near the target value have the largest magnitude, and so target values must always be provided.

Unfortunately it is difficult to say in advance what settings are optimal, or will guarantee to find the physical eigenmodes of the system. Suggested starting points for the eigenvalue settings are given below. One rule seems to be valid in general: If you use harmonic extraction and search for the mode with largest eigenvalue, you should keep the target frequency *freq=0.0*. If you don't find an eigenvalue, increase the target *growthrate*, while if you just find stable high frequency eigenmodes, decrease it.

To share some more details of our experiences, fig. 6.1–6.6 show a scan for two set of parameters. Figures 6.1–6.3 show a scan for harmonic extraction, in which the eigenvalues with largest real part where searched. The first one has *freq=0*, while *growthrate* was varied, while it is the other way round for the second figure. The last one shows for two points from the previous scans, a scan over the size of the subspace (*nr_column_vec*).

The input parameters for fig. 6.4–6.6 are based on the `STD_linear_ITG` and `simple_TEM` input files found in `doc/input`, with $R/L_{T_i} = 6$, which is near the ITG-TEM transition. Two unstable eigenmodes are therefore expected. The fixed parameters for the eigenvalue solver were

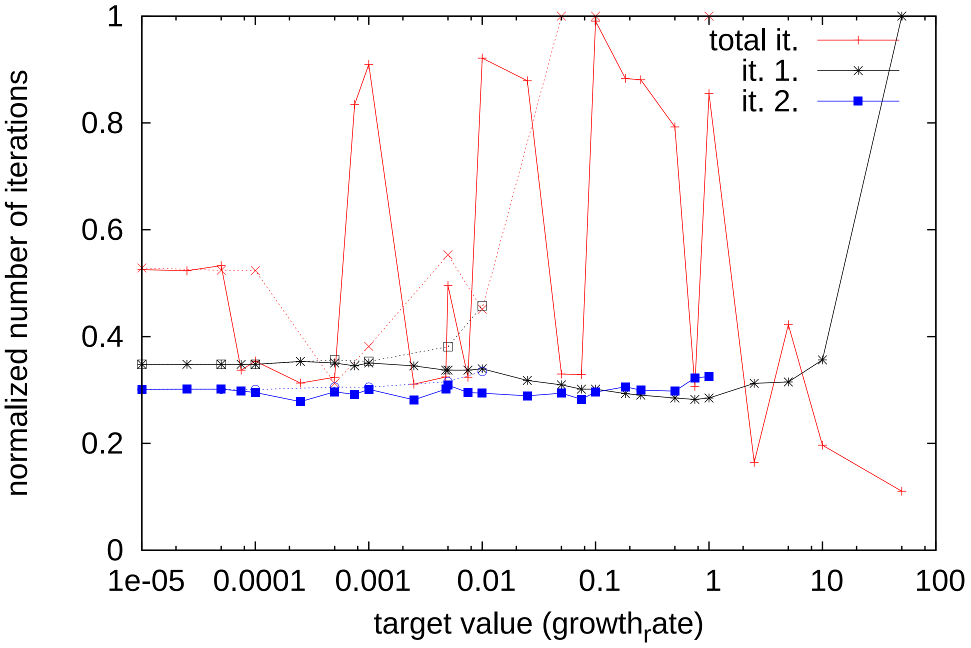


Figure 6.1: Scan over growth rate with frequency set to zero. Dashed lines depict negative values of *growthrate*. Normalization is done with the maximum number of iterations for the first/second eigenvalue/total number of iterations, respectively.

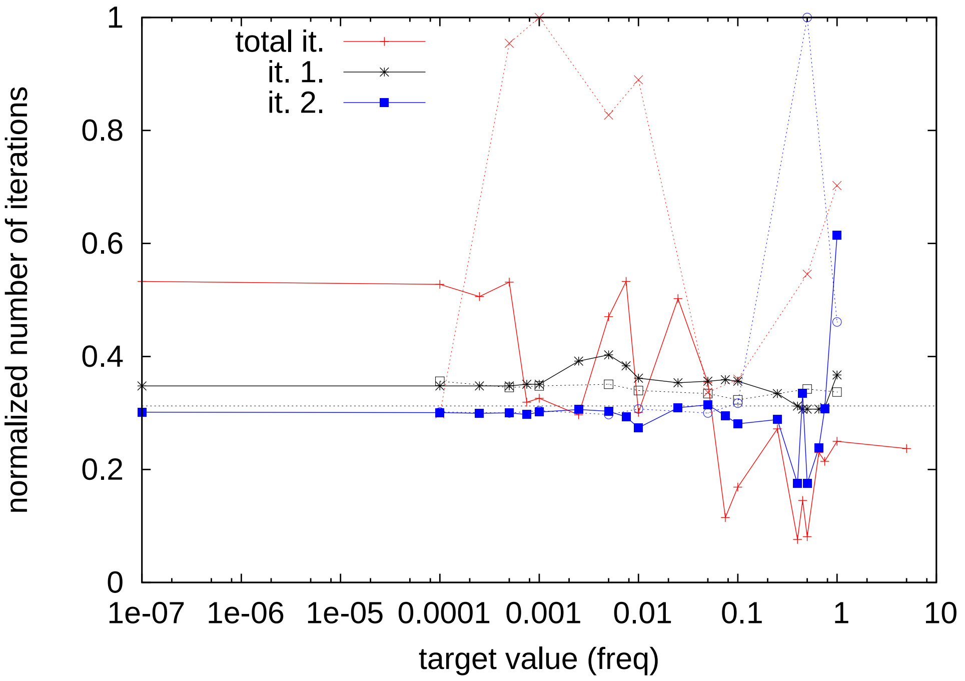


Figure 6.2: Scan over frequency with growth rate set to zero. Dashed lines depict negative values of *freq*. The point at 10^{-7} actually had zero frequency. Normalization is as for fig. 6.1. The dashed-dotted line depicts the minimum value for the total iterations of fig. 6.1 (only counting points where both unstable eigenmodes are found).

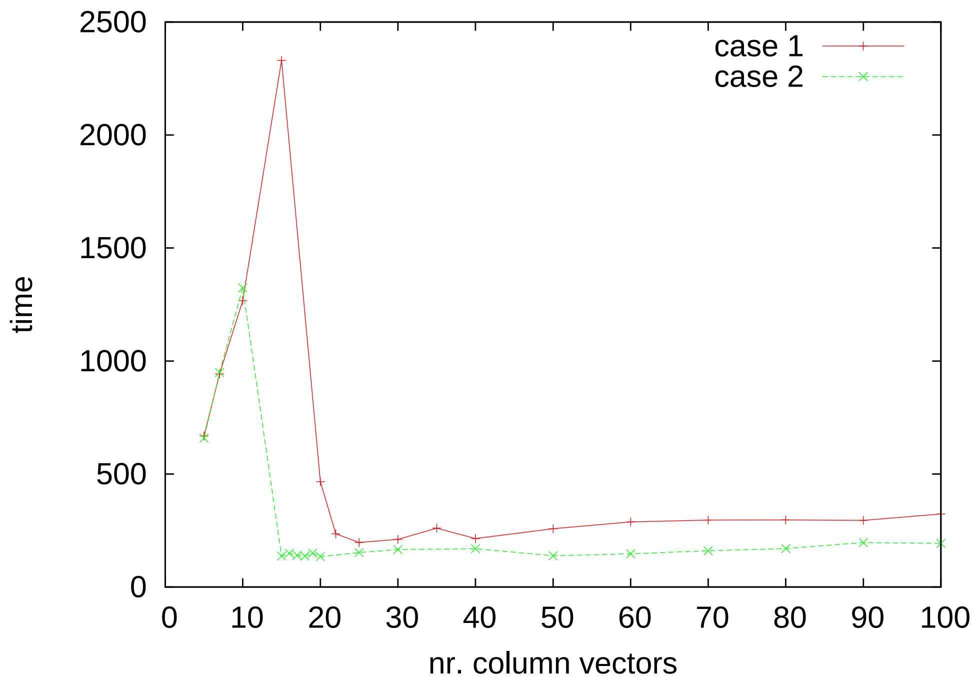


Figure 6.3: Scan over size of subspace, for one point of each of the other two scans. First case: *growthrate = 0.0*, freq = 0.45, searching for 5 eigenpairs (default value: 20). Second case: growthrate = 2.50E-3, freq = 0.0, searching for 2 eigenpairs (default value: 17).

```
max_iterations = 30000
tolerance = 1.0e-6
type_solver = 'krylovschur'
type_extraction = 'harmonic'
number_eigenvalues = 3
nr_column_vec      = 20
mat_vec_routine    = 1
comparison_routine = 2
which_eigenvalues =  1    ! largest magnitude
```

We are searching for three eigenvalues for the case that there are more unstable ones than expected. Scans were made over the target *growthrate* and *freq* for the harmonic extraction. Some remarks on the speed: If you just want to find the fastest growing eigenmode, there is not much variation. About 900-1000 iterations were needed, if these mode was found. Only for the violet points is a large difference, here 6000-8000 iterations needed to be performed. If you are also interested in the second eigenpair, then the situation is quite samilar. This is found usually 100-150 iterations after the first one in the case of fig. 6.4, while for fig. 6.5 the difference is quite often 0 or 1. The third (stable) eigenvalue then takes about 16-18 times longer. This could be avoided by setting a suitable value for max_iterations.
An exception to this is the range $growthrate = 0.75 - 0.9$ with $freq = 0$, the third eigenvalue was already found after three times the iterations needed for the second eigenvalue.

The robustness and the speed of the solver are very dependant on the initial condition vector provided. We found *finit="cosine4"* to be a good choice, but if one is doing a parameter scan, it can also be useful to restart with the output of the previous run. The batch launcher gkwnlin provides a facility (*-restart_chain*) for doing this.

The output, except for the screen output is only written if the solver finishes, which means either it finds the requested number of eigenpairs or it hits the maximum number of iterations. The output differs in some ways from the output obtained with *METHOD = 'EXP'*. The main difference is that timestep is replaced

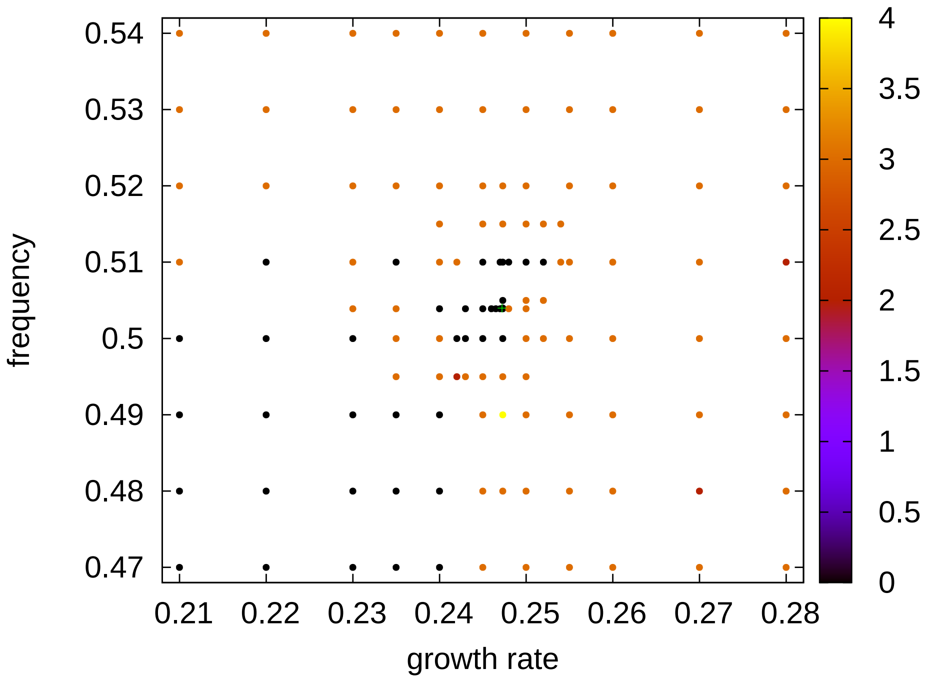


Figure 6.4: Scan for harmonic extraction and searching for eigenvalue with largest magnitude over the target values. Plus signs refer to actual found eigenvalues, crosses to 'conjugated' eigenvalues, stars to the projections on the growth rate axis. Color of the dots refers to the number of found eigenvalues. Please note, that a value of 1(or more) requires that the fastest growing eigenmode was found and a value of 2(or more) requires that both fastest growing eigenmodes where found.

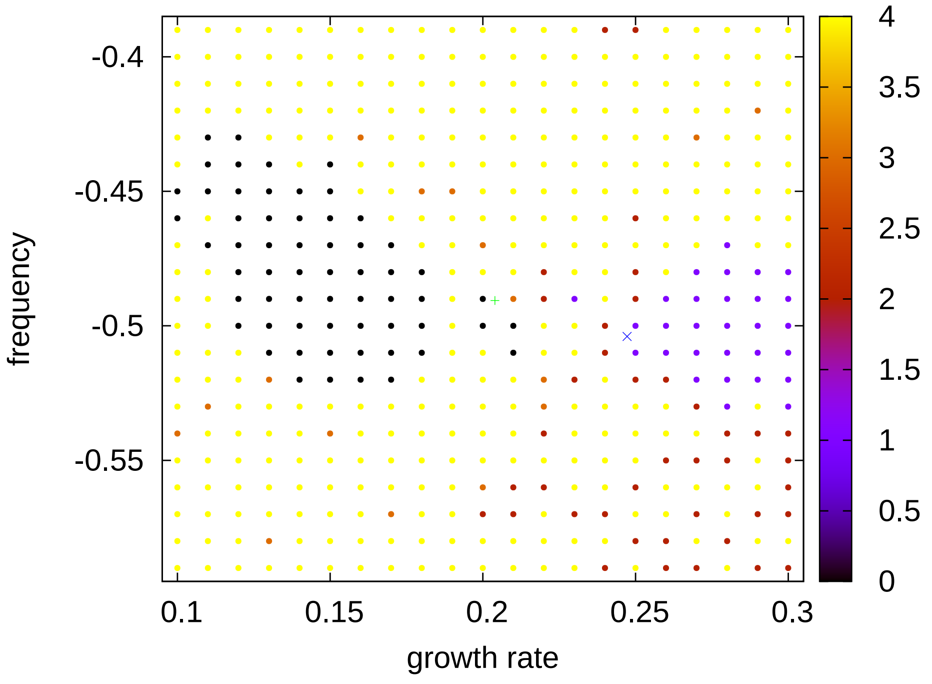
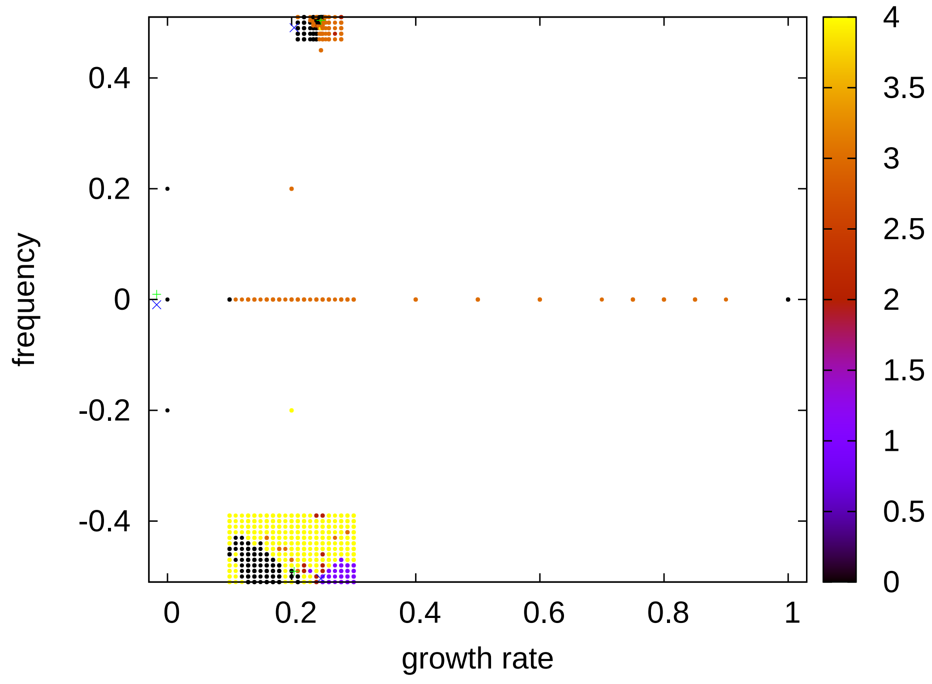


Figure 6.5: Scan for harmonic extraction and searching for eigenvalue with largest magnitude over the target values. Here is scanned around the second eigenvalue. For the meaning of the symbols please refer to the caption of fig. 6.4.



[htp!]

Figure 6.6: Scan for harmonic extraction and searching for eigenvalue with largest magnitude over the target values. Scan along *freq* = *0*, some of the other two scans can also be seen. For the meaning of the symbols please refer to the caption of fig. 6.4.

by mode in files with time dependent output. For example, the first line in `fluxes.dat` will contain the fluxes determined for the first found eigenmode, the second line those for the second eigenmode, and so on. Second, there is an output file `eigenvalues.dat` that contains the growth rate and frequency determined by SLEPC (first column acts as an index). Also you will notice that not a single `parallel.dat` or restart file is produced, but one for each found eigenmode.

If some eigenvalues are found, you should check that these belong to a reasonable, physical mode. If the eigenvalues given by the diagnostics (found in `time.dat`) and the one determined by slepc (found in `eigenvalues.dat`) differ this is a sign (not a necessary one) that the result is unphysical, but the easiest possibility is to plot the mode eigenfunction. Note also that some of the eigenvalues found could be physical while others are not. Also a warning regarding parallelization: different parallel decompositions might result in different eigenmodes being found, in particular for stable eigenmodes.

6.2 Technical details

The eigenvalue solver in GKW is mainly a interface for a library that does this job. So far the library SLEPC (which relies on PETSC) is used (and there are currently no plans to add support for/change support to other libraries).

Most parameters of the *eiv_integration* namelist act as wrapper for the corresponding parameter in SLEPC. Via switches the call to the corresponding routine with the wanted argument is done.

Note that no matrix is formed by SLEPC, as the so called matrix-free approach is used. This means instead of a matrix we just define the action of the matrix on a vector via a function, provided to SLEPC. Depending on the settings in the input file, either `exp_integration` (*mat_vec_routine* = *1*) is used to determine the result of the matrix vector multiplication or just the subroutine `calculate_rhs` (*mat_vec_routine* = *2*).

6.2.1 The actual eigenvalue problem

In general an eigenvalue problem is formulated as

$$Mx = \lambda x, \tag{6.1}$$

where A is the operator of the problem and λ is the (complex) eigenvalue. The `exp_integration` part, solves

$$f_{t+1} = f_t + dtAf_t + dtB\phi \tag{6.2}$$

$$0 = Cf_t + D\phi \tag{6.3}$$

with the state vector f, four matrices A, B, C and D and ϕ represents all the fields that may be present in the simulation. Note that `fdisi` = (f, phi). It principle it is possible to solve the last equation for ϕ and insert it into the former to get

$$f_{t+1} = ft + dtAf_t - dtBD^{-1}Cf_t. \tag{6.4}$$

The disadvantage of this approach would be, that the inversion of the matrix D and the matrix matrix multiplication most probably would destroy the sparsity structure of the complete operator. As is clear, this would make the determination of the eigenvalues much more expensive. For this reason the matrix is not explicitly formed. If the subroutine `calculate_rhs` is used instead, then the f_t term vanishes, but otherwise the computation remains unchanged, thus also in this case a shell matrix is used.

6.2.2 Transformation between eigenvalue and growth rate/frequency

The transformation between the eigenvalue computed by SLEPC and the growth rate and frequency from the code can be derived as follows. Recall again the general form of the eigenvalue problem (6.1), in comparison to this, the eigenvalue problem in GKW is

$$lf = (1 + dtN)^n f \tag{6.5}$$

where N sums up the parts from A and B and n is `naverage`. Identifying M and $(1 + dtN)^n$ results in

$$\lambda = (1 + dtN) = l \tag{6.6}$$

$$\ln \lambda = n \ln l \tag{6.7}$$

$$\frac{\ln \lambda}{ndt} = \frac{\ln l}{ndt} \tag{6.8}$$

The right hand side of these equation is the same relation as between the change in the norm and the growth rate in the code. We take this as reason to propose this as general transformation between the eigenvalue determined by slepc and the growth rate/frequency got by gkw (another reason is it works).

The transformation for the tolerance a is

$$\begin{aligned}
a_{gkw} &= \sqrt{\left(\frac{\partial \lambda_{gkw}}{\partial \lambda_{slepc}} |\lambda_{slepc}| a_{slepc}\right)^2} && (6.9)\\
&= \sqrt{\left(\frac{\partial(\log(\lambda_{slepc})/(n_{average}\Delta t))}{\partial \lambda_{slepc}} |\lambda_{slepc}| a_{slepc}\right)^2} && (6.10)\\
&= \sqrt{\left(\frac{1}{n_{average}\Delta t} a_{slepc}\right)^2} && (6.11)\\
&= \frac{a_{slepc}}{n_{average}\Delta t} && (6.12)
\end{aligned}$$

6.2.3 User defined routines

There are two points at which SLEPC uses subroutines provided by GKW. Number one is – as should be clear – the matrix vector product. As the already existing capabilities of gkw should be used, these first copy the input vector into `fdisi`, the necessary operations are performed and then the resulting new state is copied to the output vector. "Necessary operations" is so far either using `explicit_integration` or `calculate_rhs`, both from the `exp_integration` module.

Ordering the eigenvalues, is the second place where a (slepc-)user defined routine is used. These ordering functions determine which part of the spectrum is wanted. Besides of some predefined orderings, slepc also offers the possibility to let us decide, by telling which of two eigenvalues we like the most.

6.3 FAQ

What is a good starting point for the parameters? For the Ritz extraction (here for an electromagnetic case with kinetic electrons), to find the two most unstable modes you could try

```
&CONTROL
 fac_dtim_est=0.5, naverage= 1, method= 'EIV', meth= 1,
 read_file = .true., irun = 2   ! to restart from previous dominant mode
 ! using gkwnlin -restart_chain
 /
&eiv_integration  ! using Ritz extraction method, the default
 max_iterations = 90000 ! avoid looking for stable eigenmodes
 tolerance=4.0e-4
 number_eigenvalues=2
 which_eigenvalues=3    ! LARGEST REAL (most unstable modes)
 luse_initial_value = .true.
 /
&SPCGENERAL
 finit="cosine4"   ! or restart from previous run in parameter scan
 /
```

It appears that the timestep cannot go above that required for explicit time integration but that setting a very small timestep makes the convergence take longer. Setting the tolerance too small can also cause convergence to take longer, and can even prevent the correct modes from being found.

For the harmonic extraction, you can try the following.

```
 &CONTROL
 fac_dtim_est=0.5; naverage= 1, method= 'EIV', meth= 1,
 /
 &eiv_integration
 max_iterations = 50000
 tolerance = 1.0e-4
 type_solver = 'krylovschur'
 type_extraction = 'harmonic'
 number_eigenvalues = 2
 nr_column_vec      = 20
 mat_vec_routine    = 1
 comparison_routine = 2
 which_eigenvalues  = 11 ! look for eigenvalues near the target
 growthrate =  0.5000
 freq       =  0.0000
 luse_initial_value = .true.
```

```
/
&SPCGENERAL
finit="cosine4"
/
```

If you increase `number_eigenvalues` you may also have to increase `nr_column_vec`.

Does `nr_column_vec = 0` (letting 'PETSC_DECIDE') work? From the scan done over the size of the subspace (see fig. 6.3 we would expect, that it should work well for at least `number_eigenvalues = 1-2`, while for bigger values it might be not optimal.
Experience shows that even for `number_eigenvalues = 15` eigenvalues have been found.

Chapter 7

The non-spectral and global version of the code

This chapter discusses the non-spectral and global version of the code. (Work in progress)

Non-spectral here refers to the treatment of the radial direction through the use of finite difference methods, and is switched on through the switch

```
spectral_radius = .false.
```

in the control namelist. The non-spectral representation can be used both in the local limit (which we will refer to as flux-tube) as well as in the global case. Global is used as synonym for profile effects, i.e. allowing plasma and geometry parameters to be a function of the radius. A global run is selected by setting

```
flux_tube = .false.
```

in the control namelist. Naturally, global simulations can only be preformed with a non-spectral representation. Global simulations can still be δf simulations, i.e. the perturbed distribution function can be assumed small compared to the background. In this case energy is not conserved since the parallel velocity nonlinearity is neglected. Energy conserving simulations will be referred to as full-f simulations below , and can be run setting

```
lpar_vel_nl = .true.
```

in the control namelist.

Two points should be noted: First, although you can run with the parallel velocity nonlinearity in a flux-tube run, there is no point since the expansion around a local flux surface means that energy cannot be conserved (It can easily be shown that the integrated kinetic energy in the perturbed distribution is always zero in the local limit, while the integrated energy in the field must be postive. The two can then not balance each other). Therefore, global full-f, global δf, and local δf make sense, but local full-f does not, unless one wants to study some distinct property of the velocity nonlinearity. Second, care must also be taken using global δf. The turbulence will lead to a rapid profile evolution that is represented by the perturbed distribution. The condition $\delta f = \mathcal{O}[\rho_* F_M]$ is then easily violated. One way to keep δf small is through the Krook operator (see the section below), which damps the perturbed distribution on a specified timescale and acts through this damping as a source / sink of energy. But even with the use of the Krook operator the ordering is not necessarily satisfied.

7.1 Basic equation

Warning: not all is correct in this section

The equations in GKW are related to, but do not directly follow from a Lagrangian description. Further approximations to the Lagrange equations of motion are made in order to obtain a system that is easier to solve numerically. The approximations all involve neglecting higher order ρ_* terms, and since the Lagrangian from which the equations are derived itself is accurate up to terms linear in ρ_* the neglection of higher order terms is consistent with accuracy of the Lagrange description. Care has to be taken not to lose the properties that come with the Lagrange description, like conservation of energy and momentum. Below it will be shown that these properties still hold for the model equations used by GKW.

Starting point of the derivation is the Lagrangian

$$L = \sum_{sp} \int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v}\, f L_p + \sum_{sp} \int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v}\, \frac{Z_{sp}^2 e^2}{2T_{sp}}[\phi^2 - \langle\phi\rangle^2]F_M + \sigma_a \int \mathrm{d}^3\mathbf{X}\, L_{\mathrm{adiabatic}} + \int \mathrm{d}^3\mathbf{X}\, \frac{1}{2\mu_0}|\nabla A_\parallel|^2 \tag{7.1}$$

with

$$L_p = \gamma_a \dot{z}^a - H = \left(Ze\mathbf{A} + m\mathbf{u}_0 + mv_\parallel \mathbf{b} + Ze\langle A_\parallel\rangle\mathbf{b}\right)\cdot\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} - \left(\frac{1}{2}mv_\parallel^2 + \mu B + Ze\langle\phi\rangle + \frac{1}{2}mu_0^2\right), \tag{7.2}$$

where in the latter equation the species index is supressed. In the equations above the rapid gyro-motion has been removed from the Lagrangian and the indices in the Einstein summation convention are $a = 1, 2, 3, 4$ with $a = 4$ refering to the parallel velocity (Note that γ_4 is zero though). The brackets $\langle\rangle$ correspond to the gyro-average operator, and the sum with index sp is over all kinetic species. The vector potential $\mathbf{A}$ describes the (constant) background magnetic field, while the perturbed magnetic field is described by $\langle A_\parallel\rangle$.

In the case of adiabatic electrons, $\sigma_a = 1$ and the electron contribution is not included in the sum over sp (for kinetic electrons $\sigma_a = 0$. The Lagragian density associated with adiabatic electrons is

$$L_{\mathrm{adiabatic}} = -\frac{e^2 n_e}{2T_e}[\phi - \{\phi\}]^2, \tag{7.3}$$

where $\{\}$ indicate the flux surface average.

Variation towards ϕ yields the Poisson equation

$$\sum_{sp} Z_{sp} e \int \mathrm{d}^3\mathbf{v}\, \langle f\rangle^\dagger - \sigma_a \frac{n_e e^2}{T_e}(\phi - \{\phi\}) = \sum_{sp} \frac{Z_{sp}^2 e^2}{T_{sp}} \int \mathrm{d}^3\mathbf{v}\, [\phi - \langle\langle\phi\rangle\rangle^\dagger]F_{Msp} \tag{7.4}$$

and variation towards $A_\parallel$ yields Ampere's law

$$-\nabla^2 A_\parallel = \sum_{sp} Z_{sp}\mu_0 e \int \mathrm{d}^3\mathbf{v}\, \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t}\cdot\mathbf{b}\langle f\rangle^\dagger \tag{7.5}$$

GKW uses the Poisson equation and Ampere's law above, i.e. they follow directly from the Lagrange description. Approximations, however, are made in the equations of motion of the particles. The Lagrange description yields the equations of motion through the variation of the action

$$\delta S = \delta \int_{t_0}^{t_1} L_p \mathrm{d}t = \int_{t_0}^{t_1} \frac{\partial\gamma_a}{\partial z^b}\delta z_b \dot{z}^a + \gamma_a \delta\dot{z}^a - \frac{\partial H}{\partial z^b}\delta z^b \mathrm{d}t. \tag{7.6}$$

Integrating the term with $\delta\dot{z}^a$ once partial towards the time yields the equations of motion in the form

$$[\gamma_{b,a} - \gamma_{a,b}]\frac{\mathrm{d}z^b}{\mathrm{d}t} = \frac{\partial H}{\partial z^a} + \frac{\partial\gamma_a}{\partial t} \tag{7.7}$$

The equation above can be spit in the components $(1, 2, 3)$ and the fourth component as follows

$$(\nabla\times\mathbf{A}^*)\times\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} + \frac{\partial\mathbf{A}^*}{\partial v_\parallel}\frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = -\nabla H - \frac{\partial\mathbf{A}^*}{\partial t} \tag{7.8}$$

$$\frac{\partial \mathbf{A}^*}{\partial v_\parallel} \cdot \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \frac{\partial H}{\partial v_\parallel} \tag{7.9}$$

where

$$Ze\mathbf{A}^* = Ze\mathbf{A} + \mathbf{u}_0 + mv_\parallel \mathbf{b} + A_\parallel \mathbf{b} \qquad \mathbf{B}^* = \nabla \times \mathbf{A}^* \tag{7.10}$$

The equation above determine both the evolution of the gyro-centre position as well as the parallel velocity. A set of reduced equations can be derived by taking the inner product of the first equation with $\mathrm{d}\mathbf{X}/\mathrm{d}t$ to obtain

$$\left(\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot Ze \frac{\partial \mathbf{A}^*}{\partial v_\parallel} \right) \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = -\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot \nabla H - Ze \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot \frac{\partial \mathbf{A}^*}{\partial t} \tag{7.11}$$

$$Ze \frac{\partial \mathbf{A}^*}{\partial v_\parallel} \cdot \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \frac{\partial H}{\partial v_\parallel} \tag{7.12}$$

The essential observation here is that these equations will garantee energy conservation independent of the choice of $\mathrm{d}\mathbf{X}/\mathrm{d}t$. We will use in the proof below that

$$\frac{1}{J_v} \frac{\partial J_v}{\partial t} + \frac{1}{J_v} \nabla \cdot \left[J_v \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \right] + \frac{1}{J_v} \frac{\partial}{\partial v_\parallel} \left[J_v \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} \right] = 0, \tag{7.13}$$

where J_v is the Jacobian of the velocity space integration. While this condition is perhaps not under all circumstances necessary to obtain exact energy conservation, it is in our derivation an additional constraint that must be satisfied for the gyrocentre velocity. This observation allows one to make approximations to $\mathrm{d}\mathbf{X}/\mathrm{d}t$, removing higher order ρ_* corrections, without breaking energy conservation.

Combining the equations above we have

$$\frac{1}{J_v} \frac{\partial}{\partial v_\parallel} \left[J_v H \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} \right] + \frac{1}{J_v} \nabla \cdot \left[J_v H \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \right] = -Ze \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot \frac{\partial \mathbf{A}^*}{\partial t} \tag{7.14}$$

Then we split the Hamilton in a field part ($H_1 = Ze\langle\phi\rangle$) and the rest H_0 (which is time independent). To obtain

$$\frac{\mathrm{d}H_0}{\mathrm{d}t} = \frac{1}{J_v} \frac{\partial}{\partial v_\parallel} \left[J_v H_0 \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} \right] + \frac{1}{J_v} \nabla \cdot \left[J_v H_0 \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \right] = -\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot \left[\nabla H_1 + Ze \frac{\partial \mathbf{A}^*}{\partial t} \right] \tag{7.15}$$

Using furthermore that $\mathrm{d}f/\mathrm{d}t = 0$ we have

$$\frac{\mathrm{d}}{\mathrm{d}t} [H_0 f] = -\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot \left[\nabla H_1 + Ze \frac{\partial \mathbf{A}^*}{\partial t} \right] f \tag{7.16}$$

The right hand side of this equation represents the inner product of current and electric field. It is the transfer term between kinetic and field energy.

Multiplying the time derivative of the Poisson equation with ϕ we obtain

$$\int \mathrm{d}^3\mathbf{X} \int \mathrm{d}^3\mathbf{v}\, H_1 \frac{\partial f}{\partial t} = \int \mathrm{d}^3\mathbf{X} \phi \frac{\partial}{\partial t} \left[\frac{\delta H_{2E}}{\delta \phi} \right] \tag{7.17}$$

Where

$$H_{2E} = \int \mathrm{d}^3\mathbf{v}\, \frac{Z_{sp}^2 e^2}{2T_{sp}} [\phi^2 - \langle\phi\rangle^2] F_M + \sigma_a L_{adiabatic} \tag{7.18}$$

And using $\mathrm{d}f/\mathrm{d}t = 0$ the first term in the equation can be rewritten in the form

$$-\int \mathrm{d}^3\mathbf{X} \int \mathrm{d}^3\mathbf{v}\, \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot \nabla H_1\, f = \int \mathrm{d}^3\mathbf{X} \phi \frac{\partial}{\partial t} \left[\frac{\delta H_{2E}}{\delta \phi} \right] \tag{7.19}$$

The right hand side can be rewritten as

$$\int \mathrm{d}^3\mathbf{X}\phi \frac{\partial}{\partial t}\left[\frac{\delta H_{2E}}{\delta\phi}\right] = \sum_{sp}\int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v}\,\frac{Z_{sp}^2 e^2}{2T_{sp}}[\phi^2 - \langle\phi\rangle^2]F_{Msp} + \sigma_a \int \mathrm{d}^3\mathbf{X}\,\frac{n_e e^2}{2T_e}(\phi - \{\phi\})^2 \tag{7.20}$$

Finally multiplying Ampere's law with $\partial A_\|/\partial t$ and integrating over the whole of space we obtain

$$\sum_{sp} Z_{sp}e \int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v}\,\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t}\cdot\frac{\partial \mathbf{A}^*}{\partial t}f = \frac{1}{2\mu_0}\int \mathrm{d}^3\mathbf{X}\,|A_\||^2 \tag{7.21}$$

Consequently the energy conservation theorem is

$$E = \sum_{sp}\int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v}\left(\frac{1}{2}mv_\|^2 + \mu B + \frac{1}{2}mu_0^2\right)f + \sum_{sp}\int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v}\,\frac{Z_{sp}^2 e^2}{2T_{sp}}[\phi^2 - \langle\phi\rangle^2]F_{Msp} + \sigma_a\int \mathrm{d}^3\mathbf{X}\,\frac{n_e e^2}{2T_e}(\phi - \{\phi\})^2 + \frac{1}{2\mu_0} \tag{7.22}$$

where the first part is the kinetic energy while the second part is the field energy. The Noehter theorem on the original Maxwellian gives exactly the same energy theorem. Note too that no assumption has been made on the form of $\mathrm{d}\mathbf{X}/\mathrm{d}t$ other than that the equations and ?? must be satisfied.

Taking the cross product of Eq. (7.8) with the unit vector along the magnetic field ($\mathbf{b}$) one obtains the velocity of the gyro-centre

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \frac{\mathbf{B}^*}{B_{\|*}}v_\| + \frac{\mathbf{b}}{B_\|^{"}}\times\nabla H \tag{7.23}$$

where

$$B_\|^* = \mathbf{b}\cdot\mathbf{B}^* = \mathbf{b}\cdot\nabla\times\mathbf{A}^* \tag{7.24}$$

and $J_v = B_\|^*$. When used in this form $B_\|^*$ is a function of the field $\langle A_\|\rangle$, and therefore of time. Since $\mathrm{d}\mathbf{X}/\mathrm{d}t$ depends nonlinearly on $B_\|^*$ this complicates the numerical solution. At each time point the gyro-centre velocity would have to be constructed explicitly and the various terms in the equation can not be precalculated and stored in matrix format.

The approximation made in GKW is the approximation of $B_\|^*$ by B. Working out $\mathbf{B}^*$ we then obtain

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = v_\|\mathbf{b} + \frac{mv_\|^2}{ZeB}\nabla\times\mathbf{b} + \frac{mv_\|}{ZeB}\mathbf{\Omega} + \frac{1}{B}\nabla\times[v_\|\langle A_\|\rangle\mathbf{b}] + \frac{1}{B}\mathbf{b}\times\nabla H \tag{7.25}$$

with $J_v = B$ it can be shown that Eq. (7.13) is satisfied.

A few remarks should be made here. First, the approximation

$$\nabla\times\langle A_\|\rangle\mathbf{b} \approx -\mathbf{b}\times\nabla\langle A_\|\rangle \tag{7.26}$$

is problematic for exact energy conservation since in this case Eq. (7.13) is not satisfied.

7.1.1 Local limit

A brief look ahead for the local limit approximation may be in order. The local limit corresponds to a further approximation also in the gyro-centre velocity and must be separately discussed. Essentially, in the local limit all quantities are a function of the parallel coordinate (s) only. The gyro-centre velocity does not depend on the coordinates perpendicular to the field and, therefore the divergence of this velocity is different from the expression derived above. If one assumes that the drift velocity ($\mathbf{v}_D$) and ExB velocity ($\mathbf{v}_E$)do not have a component in the direction of ∇s, (i.e. $\mathbf{v}_D\cdot\nabla s = \mathbf{v}_\mathbf{E}\cdot\nabla s = 0$, which is satisfied to lowest order in ρ_* and is strictly imposed in the code when the finite ρ_* parallel derivatives are turned of) then

$$B(v_\|\mathbf{b} + \mathbf{v}_D + \mathbf{v}_E)\cdot\nabla f = \nabla\cdot[B(v_\|\mathbf{b} + \mathbf{v}_D + \mathbf{v}_E)f] \tag{7.27}$$

i.e. the gyro-centre velocity satisfies

$$\frac{1}{B}\nabla \cdot \left[B\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \right] = 0 \tag{7.28}$$

Then

$$\frac{\partial}{\partial v_\parallel}\left[\frac{\mathrm{d}v_\parallel}{\mathrm{d}t}\right] \quad \rightarrow \quad \frac{\partial}{\partial v_\parallel}\left[-\frac{1}{m}\mathbf{b}\cdot\mu\nabla B\right] = 0 \tag{7.29}$$

must be zero in order for phase space conservation to apply. The terms on the right (the trapping term) represent the only term of $\mathrm{d}v_\parallel/\mathrm{d}t$ that is kept for the perturbed distribution f. Indeed for this term the parallel velocity derivative is zero and, consequently the phase space volume is conserved in the evolution of f.

The local limit is obtained through an expansion in ρ_*, and the velocity nonlinearity is then neglected. This neglection means that energy is not strictly conserved. In fact, in the local limit with periodic boundary conditions energy conservation is problematic since the total field energy is always positive, whereas the total kinetic energy

$$\int \mathrm{d}^3\mathbf{X} \int \mathrm{d}^3\mathbf{v} \left[\frac{1}{2}mv_\parallel^2 + \mu B\right] f \tag{7.30}$$

integrates to zero for all Fourier modes except the mode that has a zero wave vector in both perpendicular directions (below this mode is refered to as 0,0 mode). The latter Fourier mode is usually not kept in the evolution equation, and it is then not possible to conserve the energy since the positive field energy can not be balanced with a negative total kinetic energy.

Energy conservation however can be obtained if the 0,0 mode is kept in the evolution equations, provided the velocity nonlinearity is implemented as

$$\frac{\partial f}{\partial t} \stackrel{+}{=} -\frac{\partial}{\partial v_\parallel}\left[\frac{\mathrm{d}v_\parallel}{\mathrm{d}t} f\right]. \tag{7.31}$$

This form ensures phase space conservation in the local limit where the divergence of the gyro-centre velocity is zero. In the equation above the change in the parallel velocity is considered only through the electro-magnetic field. It can easily be shown that when integrated over the whole computational domain the term only contributes to the 0,0 mode. The 0,0 mode has no spatial derivatives perpendicular to the field, and except through the velocity nonlinearity it does not couple to any other mode. The 0,0 mode therefore only acts as a reservoir for the kinetic energy essuring that the balance equation for the energy holds.

7.1.2 Toroidal momentum conservation

The equations of motion (Eq. (7.7) can be written in the form

$$\frac{\mathrm{d}\gamma_a}{\mathrm{d}t} = -\frac{\partial H}{\partial z^a} - \gamma_{b,a}\frac{\mathrm{d}z^b}{\mathrm{d}t} \tag{7.32}$$

In the case of a symmetry in coordinate z^a we have

$$\frac{\partial H}{\partial z^a} = 0 \qquad \gamma_{b,a} = 0 \tag{7.33}$$

and therefore

$$\frac{\mathrm{d}\gamma_a}{\mathrm{d}t} = 0 \tag{7.34}$$

This represents a conservation equation. The case of interest here is the toroidal symmetry of the tokamak, and the consequent conservation of toroidal angular momentum. The equations here are slightly different since both $\langle \phi \rangle$ and $\langle A_\parallel \rangle$ are a function of the toroidal angle. Choosing the toroidal angle φ as coordinate one obtains

$$\frac{\mathrm{d}\gamma_\varphi}{\mathrm{d}t} = -\frac{\partial H_1}{\partial \varphi} - Ze\frac{\partial \langle A_\parallel \rangle}{\partial \varphi} v_\parallel \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \tag{7.35}$$

Multiplying this equation with f and integrating over the whole phase space (using $\mathrm{d}f/\mathrm{d}t = 0$ we obtain

$$\frac{\partial}{\partial t}\int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v}\, \gamma_\varphi f = -\int \mathrm{d}^3\mathbf{X}\mathrm{d}^3\mathbf{v} \left[\frac{\partial H_1}{\partial \varphi} + Ze\frac{\partial \langle A_\parallel \rangle}{\partial \varphi} v_\parallel\right] f \tag{7.36}$$

The right hand side of this equation can be shown to vanish

7.2 Normalization

In the local version of GKW the velocity coordinates are normalized to the thermal velocity of the considered species. This has the advantage that the Maxwell can be represented with the same accuracy for each of the species even if the temperatures of the species are very different. When the temperature is a function of the radial coordinate, however, the thermal velocity is no longer a constant. Using the local thermal velocity for normalization would have the advantage that the Maxwell is represented with the same relative resolution at each radial point, but would make the normalization constant a function of radius, and would lead to cross terms in all radial derivatives. A choice has therefore been made to use a fixed temperature, often the temperature at one particular radial location, for the normalization. This has the advantage that it does not change the structure of the equations, but also has disadvantages. In the global case the grid must be, both set up such that it is big enough to represent the maximum temperature ($T_{\max}$) in the domain, as well as have sufficient resolution to resolve the distribution function of the smallest temprature ($T_{\min}$). Because the velocity scales as the square root of the temperature this increases the computational cost by a factor $T_{\max}/T_{\min}$ over a local simulation.

In the text below all normalized quantities have an index N. A symbol like T_G therefore has a physical unit. Naturally, the code uses only normalized quantities, and physical units can only be obtained through the reference quatities

$$T_{\rm ref}, \qquad n_{\rm ref}, \qquad m_{\rm ref}, \qquad R_{\rm ref}, \qquad B_{\rm ref}$$

the reference temperature, density, mass, major radius and magnetic field. These quantities are never explicitly specified. The numerical solution is valid for any choice of these quantities. Note though that the collision frequency leads to an additional constraint between the quantities mentioned above.

The temperature used for the normalization of the velocity coordinates is T_G. T_G is not a function of the radius but, unlike $T_{\rm ref}$, is different for each species. The thermal velocity used for normalization then is

$$w = \sqrt{\frac{2T_G}{m}}$$

Where m is the particle mass. Using this normalization the velocity space coordinates are

$$v_\parallel = w v_{\parallel N} \qquad \mu = \frac{w^2}{B_{\rm ref}}\mu_N$$

(Note that μ in the code is defined without the mass). Besides the temperature a similar normalization is used for the density. Each of the species is normalized to the 'grid' density n_G, with

$$n_{GN} = n_G/n_{\rm ref} \qquad n_N = n/n_{\rm ref}$$

The distribution functions are normalized as follows

$$f = \rho_* \frac{n_G}{w^3} f_N \qquad F_M = \frac{n_G}{w^3} F_{MN}$$

Temperatures and masses are normalized with $T_{\rm ref}$ and $m_{\rm ref}$

$$T_N = T/T_{\rm ref} \qquad T_{GN} = T_G/T_{\rm ref} \qquad m_N = m/m_{\rm ref}$$

and since $T_{\rm ref}$ and $m_{\rm ref}$ define the reference thermal velocity

$$T_{\rm ref} = \frac{1}{2} m_{\rm ref} v_{thref}^2$$

we have

$$w_N = \frac{w}{v_{thref}} = \sqrt{\frac{T_{GN}}{m_N}}$$

Note that only the velocity space coordinates are normalized with w. Species independent quantities, like time, rotation velocity and normalized Larmor radius are normalized using $v_{\rm thref}$

$$t = \frac{R_{\rm reff}}{v_{\rm thref}} t_N \qquad \Omega = v_{\rm thref}\Omega_N/R_{\rm reff}$$

$$\rho_{\rm ref} = \frac{m_{\rm ref} v_{thref}}{eB_{\rm ref}} \qquad \rho_* = \rho_{\rm ref}/R_{ref}$$

The fields are also normalized with $T_{\rm ref}$

$$\phi = \rho_* \frac{T_{\rm ref}}{e}\phi_N \qquad A_\parallel = B_{\rm ref} R_{\rm ref} \rho_*^2 A_{\parallel N}$$

Finally

$$R = R_{\rm ref} R_N, \qquad B = B_{\rm ref} B_N, \qquad m = m_{\rm reff} m_N, \qquad \Phi = \frac{T_{\rm ref}}{e}\Phi_N$$

$$\chi = \frac{T_{\rm ref}}{e}\rho_* \chi_N, \qquad \mathcal{E}_\Omega = \mathcal{E}_{\Omega N} T_G$$

With this normalization the Maxwell can be written as

$$F_{MN} = \frac{n_N/n_{GN}}{\pi^{3/2}(T_N/T_{GN})^{3/2}} \exp\left[-\frac{(v_{\parallel N} - \sqrt{m_N/T_{GN}} R_N B_{tN} \omega_{\phi N}/B_N)^2 + 2\mu_N B_N + \mathcal{E}_{\Omega N}}{T_N/T_{GN}} \right]$$

In the above equation, only n_N, T_N, R_N and B_N are functions of radius. It is to be noted that the 'old' flux-tube normalization is included in the above normalization if one sets $T_G = T$. This choice is made by the code when choosing `flux_tube = .true.`. Furthermore, it should be noted that the frame rotation (Ω) is not a function of the radius. This is more than just a choice. If the frame rotation is a function of the radius, two stationary points in the co-moving frame could shear apart in the laboratory frame. In short, the metric tensor would be a function of time, which is highly undesirable. The co-moving frame description of the rotation is, therefore, useful for the flux tube case, but less so for the global case. The use of Ω in the global case is allowed, but the implementation is not entirely consistent (the $\mathcal{E}_\Omega$ quantity, for instance, is not a function of the radius). The user is therefore strongly discouraged from using the plasma rotation Ω in the global case. Momentum fluxes due to the u' however can be calculated. In the future a rotation profile will be implemented through a shift of the perturbed distribution, rather than a coordinate transformation.

With the new normalization the equations of motion are (below all quantities are normalized, i.e. w in these equations refers to w_N, etc.)

$$\frac{\partial g}{\partial t} = \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV} + \mathrm{V} + \mathrm{VI} + \mathrm{VII} + \mathrm{VIII}, \tag{7.37}$$

with

$$\mathrm{I} = -v_\parallel \mathbf{b}\cdot\nabla f \to -wv_\parallel \mathcal{F}\frac{\partial f}{\partial s}, \tag{7.38}$$

$$\mathrm{II} = -\mathbf{v}_D\cdot\nabla f \to$$

$$-\frac{\rho_*}{Z}\left[T_G E_D \mathcal{D}^\alpha + T_G v_\parallel^2 \beta' \mathcal{E}^{\psi\alpha} + 2mwv_\parallel \Omega \mathcal{H}^\alpha + m\Omega^2 I^\alpha + Z\mathcal{E}^{\beta\alpha}\frac{\partial \Phi}{\partial x_\beta}\right]\frac{\partial f}{\partial x_\alpha} \tag{7.39}$$

$$\mathrm{III} = -\mathbf{v}_\chi\cdot\nabla g \rightharpoonup -\rho_*^2 \frac{\partial \chi}{\partial x_\beta}\mathcal{E}^{\beta\alpha}\frac{\partial g}{\partial x_\alpha} \tag{7.40}$$

$$\mathrm{IV} = +\frac{\mathbf{b}}{m}\cdot(\mu\nabla B + \nabla\mathcal{E}_\Omega)\frac{\partial f}{\partial v_\parallel} \to w\left(\mu B\mathcal{G} + \frac{T}{2T_G}\mathcal{F}\frac{\partial \mathcal{E}_\Omega}{\partial s}\right)\frac{\partial f}{\partial v_\parallel}, \tag{7.41}$$

$$\mathrm{V} = -\mathbf{v}_\chi\cdot\nabla F_M \to \rho_*\frac{\partial\chi}{\partial x_\alpha}\mathcal{E}^{\alpha\psi}\left[\frac{1}{L_n} + E_T\frac{1}{L_T} + \left(\frac{2mwv_\parallel}{T}\frac{RB_t}{B} + \frac{2m\Omega}{T}\mathcal{J}\right)u' + \frac{m\Omega^2}{T}\mathcal{L}\right]F_M \tag{7.42}$$

$$\mathrm{VI} = -\mathbf{v}_D\cdot\nabla F_M \to \frac{1}{Z}\left[T_G E_D \mathcal{D}^\psi + 2mwv_\parallel\Omega\mathcal{H}^\psi + m\Omega^2 I^\psi + Z\mathcal{E}^{s\psi}\frac{\partial\Phi}{\partial s}\right] \tag{7.43}$$

$$\times\left[\frac{1}{L_n} + E_T\frac{1}{L_T} + \left(\frac{2mwv_\parallel}{T}\frac{RB_t}{B} + \frac{2m\Omega}{T}\mathcal{J}\right)u' + \frac{m\Omega^2}{T}\mathcal{L}\right]F_M, \tag{7.44}$$

$$\mathrm{VII} = -\frac{Ze}{T}v_\parallel\mathbf{b}\cdot\nabla\langle\phi\rangle F_M \to -\frac{Z}{T}wv_\parallel\mathcal{F}\frac{\partial\langle\phi\rangle}{\partial s}F_M, \tag{7.45}$$

$$\mathrm{VIII} = -\frac{Ze}{T}\mathbf{v}_D\cdot\nabla\langle\phi\rangle F_M \to$$

$$-\frac{\rho_*}{T}\left[T_G E_D\mathcal{D}^\alpha + T_G\beta' v_\parallel^2\mathcal{E}^{\psi\alpha} + 2mwv_\parallel\Omega\mathcal{H}^\alpha + m\Omega^2\mathcal{I}^\alpha + Z\mathcal{E}^{\beta\alpha}\frac{\partial\Phi}{\partial x_\beta}\right]\frac{\partial\langle\phi\rangle}{\partial x_\alpha}F_M$$

(7.46)

where

$$\chi = \langle\phi\rangle - 2wv_\parallel\langle A_\parallel\rangle \tag{7.47}$$

and

$$g = f + \frac{2Z}{T}wv_\parallel\langle A_\parallel\rangle F_M, \tag{7.48}$$

$$E_D = v_\parallel^2 + \mu B, \qquad E_T = \frac{T}{T_G}\left[v_\parallel^2 + 2\mu B + \mathcal{E}_\Omega\right] - \frac{3}{2}. \tag{7.49}$$

$$\frac{1}{L_N} \equiv -\frac{1}{n}\frac{\partial n}{\partial\psi} \qquad \frac{1}{L_T} \equiv -\frac{1}{T}\frac{\partial T}{\partial\psi} \qquad u' \equiv -\frac{\partial\omega_\phi}{\partial\psi} \tag{7.50}$$

The equations above apply to each of the species individually.

In the equations the spatial derivatives of the perturbed distribution function appear, and the Einstein summation convention has been used. For all directions that are represented by the spectral representation we have

$$\rho_*\frac{\partial\hat{f}}{\partial x_\alpha} \to \mathrm{i}k_\alpha\hat{f} \tag{7.51}$$

The sum over α also includes $\alpha = s$. These terms are, however, ρ_* smaller than the derivatives perpendicular to the magnetic field and are usually neglected. They have been studied in connection with momentum transport in Ref. [80], and must be explicitly switched on through the input file.

7.3 The velocity nonlinearity

The velocity nonlinearity is neglected in the δf formalism since it is one order smaller in the normalized Larmor radius (ρ_*) compared with the leading order terms. The correct equation for the velocity nonlinearity can be obtained by considering the equation for the evolution of the parallel velocity (Here without rotation and only electro-static)

$$m v_\parallel \frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = -\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \cdot [Ze\nabla\langle\phi\rangle + \mu\nabla B] \tag{7.52}$$

In the δf formalism the terms that are proportional to both the field (ϕ) as well as the perturbed distribution are neglected. Note that these terms do include the term due to $\mathbf{v}_E \cdot \mu\nabla B$, which cancels $\mathbf{v}_{\nabla B} \cdot Ze\nabla\langle\phi\rangle$. The remaining terms are

$$\frac{\mathrm{d}v_\parallel}{\mathrm{d}t} = -\frac{Ze}{m v_\parallel}[v_\parallel \mathbf{b} + \mathbf{v}_{D-}] \cdot \nabla\langle\phi\rangle \tag{7.53}$$

where $\mathbf{v}_{D-}$ is the drift velocity minus the grad-B drift.

The parallel velocity nonlinearity therefore adds an additional term to the gyro-kinetic equation in the form

$$\frac{\partial f}{\partial t} \overset{+}{=} \frac{Ze}{m v_\parallel}[v_\parallel \mathbf{b} + \mathbf{v}_{D-}] \cdot \nabla\langle\phi\rangle \frac{\partial f}{\partial v_\parallel} \tag{7.54}$$

Normalizing this contribution gives for the parallel motion and the drift, respectively

$$\frac{\partial f_N}{\partial t_N} \overset{+}{=} \frac{Z\rho_*}{2\sqrt{T_{GN} m_N}} \mathcal{F} \frac{\partial\langle\phi_N\rangle}{\partial s} \frac{\partial f_N}{\partial v_{\parallel N}} \tag{7.55}$$

$$\frac{\partial f_N}{\partial t_N} \overset{+}{=} \frac{\rho_*^2}{2}\left[v_{\parallel N}\mathcal{D}^\alpha + v_{\parallel N}\beta'\mathcal{E}^{1\alpha} + 2H^\alpha\sqrt{\frac{m_N}{T_{GN}}}\Omega_N\right] \frac{\partial\langle\phi_N\rangle}{\partial x^\alpha} \frac{\partial f_N}{\partial v_{\parallel N}} \tag{7.56}$$

(here the Coriolis drift is kept which is somewhat inconsistent with the fact that it has been neglected in the equation above, but it is work in progress). The drift terms involve derivatives in the perpendicular plane that scale as $1/\rho_*$. Therefore, both terms are of the order ρ_* and are usually neglected.

As discussed in section 7.1.1, in the case of the local limit the velocity nonlinearity must be implemented as

$$\frac{\partial f}{\partial t} \overset{+}{=} \frac{\partial}{\partial v_\parallel}\left[\frac{Ze}{m v_\parallel}[v_\parallel \mathbf{b} + \mathbf{v}_{D-}] \cdot \nabla\langle\phi\rangle f\right] \tag{7.57}$$

in order to assure phase space convervation. In the latter casse an additional term appears in Eq. (7.56) which must be implemented as

$$\begin{aligned}\frac{\partial f_N}{\partial t_N} \overset{+}{=}& \frac{\rho_*^2}{2}\left[v_{\parallel N}\mathcal{D}^\alpha + v_{\parallel N}\beta'\mathcal{E}^{1\alpha} + 2H^\alpha\sqrt{\frac{m_N}{T_{GN}}}\Omega_N\right] \frac{\partial\langle\phi_N\rangle}{\partial x^\alpha} \frac{\partial f_N}{\partial v_{\parallel N}} \\ &+\frac{\rho_*^2}{2}\left[\mathcal{D}^\alpha + \beta'\mathcal{E}^{1\alpha}\right]\frac{\partial\langle\phi\rangle}{\partial x^\alpha} f\end{aligned} \tag{7.58}$$

7.4 Krook operator

In a global run the profiles relax. Without any sources or sinks this leads to a decay of the turbulence since it is no longer driven. One of the methods to obtain a 'stationary' turbulence run is to use the krook operator. The operator is implemented to conserve density and parallel velocity, but not energy. Below the krook option 1 is described.

First the density moment is calculated and stored in the array sss

$$\text{sss} = \tilde{n} = \int f \, \mathrm{d}^3\mathbf{x}$$

Then this moment is used in the calculation of the damping term

$$\left.\frac{\partial f(v_\parallel,\mu)}{\partial t}\right|_K = -\gamma_K \left[\frac{1}{2}[f(v_\parallel, mu) + f(-v_\parallel,\mu)] - \tilde{n} F_M(v_\parallel,\mu)\right]$$

where the Maxwell is here defined such that it satisfies

$$\int F_M \, \mathrm{d}^3\mathbf{v} = 1 \qquad \int \frac{1}{2} m v^2 F_M \, \mathrm{d}^3\mathbf{v} = \frac{3}{2} T_0$$

Integrating the equation over velocity space yields for the zero moment

$$\left.\frac{\partial}{\partial t}\right|_K \int f \, \mathrm{d}^3\mathbf{v} = 0$$

i.e. density is locally conserved. Taking the first parallel velocity moment

$$\left.\frac{\partial}{\partial t}\right|_K \int v_\parallel f \, \mathrm{d}^3\mathbf{v} = 0$$

i.e. also the parallel momentum is locally conserved. Of course, the energy is not

$$\left.\frac{\partial}{\partial t}\right|_K \int \frac{1}{2} m v^2 f \mathrm{d}^3\mathbf{v} = -\gamma_K \int \left[\frac{1}{2} m v^2 - \frac{3}{2} T_0\right] f \, \mathrm{d}^3\mathbf{v}$$

Since density is conserved the perturbed distribution can not be uniformly (in velocity space) damped. The expression above makes clear that the distribution function is reduced for energies more than the thermal energy of the bulk, whereas it is growing for smaller energies.

7.5 Profile functions in GKW

In the global version one can choose various analytic profiles through the switches '*dens_prof_type*' and '*temp_prof_type*' in the *SPECIES* namelist. Allowed are at present the switches '*const*', '*cosh2*', '*tanh*', '*exp_tanh*', '*orb*', '*orb3*', '*exp_poly3*' and '*exp_poly6*'. The parameters needed to calculate the profile are given in the arrays 'dens_prof_coef' and 'temp_prof_coef'. At present a maximum of five parameters is available.

Below the quantity G stands for either density or temperature, while

$$G' = \frac{1}{G}\frac{\mathrm{d}G}{\mathrm{d}\psi}$$

Besides the profiles of density and temperature one can also specify the q profile. To select a specific profile one sets the *prof_type* option in the *GEOM* namelist. The magnetic shear is calculated consistently with the profile specified. Valid options are at present: '*parabolic*', '*parabolic2*', '*orb*', '*wesson*', '*rexp*', '*mishchenko*'. The parameters needed in the analytic profile are given to the code through the *qprof_coef* array (also in the namelist GEOM)

7.5.1 Density/Temperature profile option: cosh2

$G_0 = \text{dens\_prof\_coef}(1)$	the density / temperature at $x = X_0$
$R/L_G = \text{dens\_prof\_coef}(2)$	gradient length R/L_G at $x = X_0$
$X_0 = \text{dens\_prof\_coef}(3)$	Radial location of the maximum gradient
$w = \text{dens\_prof\_coef}(4)$	the normalized width of the gradient profile $\Delta x/R_0$

$\Delta = \text{dens\_prof\_coef}(5)$	Width that determines the rise of the R/L_T profile

$$
\begin{aligned}
X &= \min(\max(X_0 - w/2, \psi), X_0 + w/2.) \\
G &= G_0 \exp\left[-\frac{R}{L_G}(X - X_0) + \frac{R}{L_G}\Delta \tanh\left(\frac{X - X_0 + w/2}{\Delta}\right) + \frac{R}{L_G}\Delta \tanh\left(\frac{X - X_0 - w/2}{\Delta}\right)\right] \\
G' &= \frac{R}{L_G}\left(1 - \frac{1}{\cosh^2((X - X_0 + w/2)/\Delta)} - \frac{1}{\cosh^2((X - X_0 - w/2)/\Delta)}\right) \\
G_{\text{norm}} &= G_0
\end{aligned}
$$

7.5.2 Density/Temperature profile option: 'const'

$G_0 = \text{dens\_prof\_coef}(1)$	the density / temperature

$$
\begin{aligned}
G &= G_0 \\
G' &= 0 \\
G_{\text{norm}} &= G_0
\end{aligned}
$$

7.5.3 Density/Temperature profile option: 'exp_tanh'

$G_0 = \text{dens\_prof\_coef}(1)$	the density / temperature at $x = X_0$
$R/L_G = \text{dens\_prof\_coef}(2)$	gradient length R/L_G at $x = X_0$
$X_0 = \text{dens\_prof\_coef}(3)$	Radial location of the maximum gradient
$w = \text{dens\_prof\_coef}(4)$	the normalized width of the gradient profile $\Delta x/R_0$

$$
\begin{aligned}
G &= G_0 \exp\left(-\frac{R}{L_G} w \tanh\left(\frac{\psi - X_0}{w}\right)\right) \\
G' &= \frac{R}{L_G}\frac{1}{\cosh^2((\psi - X_0)/w)} \\
G_{norm} &= G_0
\end{aligned}
$$

7.5.4 Density/Temperature profile option: exp_poly3

Results in a quadratic profile of the gradient length. Limiting cases of linear (for a = 0) and constant (for a=0 and b=0) are included. Some care has to be taken to get physical profiles. To get a gradient length that is positive over the whole range, the inequality $b^2 < 4a$ has to be fullfilled.

$G_0 = \text{dens\_prof\_coef}(1)$	the density / temperature at $x = X_0$
$R/L_G = \text{dens\_prof\_coef}(2)$	gradient length R/L_G at $x = X_0$
$X_0 = \text{dens\_prof\_coef}(3)$	Radial location of the maximum gradient
$a = \text{dens\_prof\_coef}(4)$	constant for the quadratic term
$b = \text{dens\_prof\_coef}(5)$	constant for the linear term

$$G = G_0 \exp\left[-\frac{R}{L_G}\left(\frac{a}{3}(\psi - X_0)^3 + \frac{b}{2}(\psi - X_0)^2 + (\psi - X_0)\right)\right]$$
$$G' = \frac{R}{L_G}\left[a(\psi - X_0)^2 + b(\psi - X_0) + 1\right]$$
$$G_{\rm norm} = G_0$$

7.5.5 Density/Temperature profile option: exp_poly6

Results in a fifth and third order term as well as a constant for the gradient length.

$G_0 =$ dens_prof_coef(1)	the density / temperature at $x = X_0$
$R/L_G =$ dens_prof_coef(2)	gradient length R/L_G at $x = X_0$
$X_0 =$ dens_prof_coef(3)	Radial location of the maximum gradient
$a =$ dens_prof_coef(4)	constant for the fifth order term
$b =$ dens_prof_coef(5)	constant for the third order term

$$G = G_0 \exp\left[-\frac{R}{L_G}\left(\frac{a}{6}((\psi - X_0)^6 + \frac{b}{4}((\psi - X_0)^4 + (\psi - X_0)\right)\right]$$
$$G' = \frac{R}{L_G}\left[a((\psi - X_0)^5 + b(\psi - X_0)^3 + 1\right]$$
$$G_{\rm norm} = G_0$$

7.5.6 Density/Temperature profile option : orb

This produces profiles equivalent to one of the options used in ORB5 (NEMORB). The calculation is a little more involved because these orb profiles are defined as function of a different radial coordinate.

$G_0 =$ dens_prof_coef(1)	the density / temperature at $x = X_0$
$R/L_G =$ dens_prof_coef(2)	gradient length R/L_G at $x = X_0$
$X_0 =$ dens_prof_coef(3)	Radial location of the maximum gradient
$w =$ dens_prof_coef(4)	Width of the profile
$X_e =$ dens_prof_coef(5)	minor radius of the plasma edge $X_e = a/R_0$

The values of $\bar{q}$ as used in ORB at the edge and in the centre
$$q_a = q(a)\sqrt{1 - X_e^2}$$
$$q_0 = q(0)$$
Rescale the gradient length to have exactly the input value at the s_0 coordinate used in ORB
$$\frac{R}{L_{G*}} = \frac{R/L_G}{1 - 1/\cosh^2(X_0/w)}$$
Calculate the s coordinate used in ORB
$$s = \sqrt{\left(\frac{\log(1 + (q_a - q_0)\psi^2/(q_0 X_e^2))}{\log(1 + (q_a - q_0)/q_0)}\right)}$$
$$D(X) = \exp\left(\frac{R}{L_{G*}}\frac{s^2}{\cosh(x_0/w)^2} - 2\frac{R}{L_{G*}} sw \tanh\left(\frac{s - X_0}{w}\right)\right)$$
$$G(\psi) = G_0 D(\psi)/D(X_0)$$
$$G' = \frac{R}{L_{G*}}\left(\frac{1}{\cosh^2((s - X_0)/w)} - \frac{1}{\cosh^2(x_0/w)}\right)$$
$$G_{\rm norm} = G_0$$

7.5.7 Density/Temperature profile option: 'orb3'

One of the profiles used in ORB (NEMORB)

$G_0 =$	dens_prof_coef(1)	the density / temperature at $x = X_0$
$R/L_G =$	dens_prof_coef(2)	gradient length R/L_G at $x = X_0$
$X_0 =$	dens_prof_coef(3)	Radial location of the maximum gradient
$w =$	dens_prof_coef(4)	Width of the profile
$\Delta =$	dens_prof_coef(5)	width that determines the rise of the profile

$$X = \min(\max(\psi, X_0 - w/2), x_0 + w/2)$$

$$G = G_0 \exp\left(-\frac{1}{2}\frac{R}{L_G}\Delta \log\left(\frac{\cosh((X - X_0 + w)/\Delta)}{\cosh((X - X_0 - w)/\Delta)}\right)\right)$$

$$G' = \frac{1}{2}\frac{R}{L_G}\left(\tanh\left(\frac{X - X_0 + w}{\Delta}\right) - \tanh\left(\frac{X - X_0 - w}{\Delta}\right)\right)$$

$$G_{\rm norm} = G_0$$

7.5.8 Density/Temperature profile option: 'tanh'

The tanh profile

$G_0 =$	dens_prof_coef(1)	the density / temperature at $x = X_0$
$R/L_G =$	dens_prof_coef(2)	gradient length R/L_G at $x = X_0$
$X_0 =$	dens_prof_coef(3)	Radial location of the maximum gradient
$w =$	dens_prof_coef(4)	Width of the profile
$\Delta =$	dens_prof_coef(5)	width that determines the rise of the profile

$$G = G_0 \left[1. - \frac{\Delta}{2}\frac{R}{L_G}\left(\log\left(\cosh\left(\frac{\psi - X_0 + w/2}{\Delta}\right)\right) - \log\left(\cosh\left(\frac{\psi - X_0 - w/2}{\Delta}\right)\right)\right)\right]$$

$$G' = \frac{1}{2}\frac{R}{L_G}\left(\tanh\left(\frac{\psi - X_0 + w/2)}{\Delta}\right) - \tanh\left(\frac{\psi - X_0 - w/2}{\Delta}\right)\right)$$

The norm is the value of the profile at X_0, not the G_0 value

$$G_{\rm norm} = G(X_0)$$

7.5.9 q-profile option: parabolic

Parabolic q-profile

$q_0 =$	qprof_coef(1)	q value at the axis
$a =$	qprof_coef(2)	coefficient for the quadratic term

$$q = q_0 + a\psi^2$$

with $\psi = r/R_0$ in circular geometry.

$$\hat{s} = \frac{2a\psi^2}{q_0 + a\psi^2}$$

7.5.10 q-profile option: parabolic2

Second degree polynomial

$q_0 =$ qprof_coef(1)	q value at the axis
$a =$ qprof_coef(2)	coefficient for the linear term
$b =$ qprof_coef(3)	coefficient for the quadratic term

$$
\begin{aligned}
q &= q_0 + a\psi + b\psi^2 \\
\hat{s} &= \frac{a\psi + 2b\psi^2}{/q}
\end{aligned}
$$

7.5.11 q-profile option: orb

For circular geometry one of the options in ORB is

$q_0 =$ qprof_coef(1)	q value at the axis
$a =$ qprof_coef(2)	coefficient for the quadratic term

$$
\begin{aligned}
q &= \frac{q_0 + a\psi^2}{\sqrt{1-\psi^2}} \\
\hat{s} &= \frac{2a\psi^2}{q_0 + a\psi^2} - \frac{\psi^2}{1-\psi^2}
\end{aligned}
$$

7.5.12 q-profile option: wesson

The q-profile used in Wesson. For this profile $qprof_coef(1)\psi^2 < 1$ has to hold over the entire profile, otherwise the term in the logarithm gets negative. If this happens, the simulation will abort.

$D_w =$ qprof_coef(1)	The aspect ratio squared$(R/a)^2$
$\nu =$ qprof_coef(2)	Exponent of the q-profile.
$q_0 =$ qprof_coef(2)	q value on the axis

$$
\begin{aligned}
q &= \frac{q_0 \psi^2 D_w}{1 - \exp((\nu+1)\log(1 - D_w\psi^2))} \\
\hat{s} &= 2\left(1 - \frac{q(\nu+1)}{q_0}\exp\left(\nu \log(1 - D_w\psi^2)\right)\right)
\end{aligned}
$$

7.5.13 q-profile option: rexp

Exponential q-profile, with a cut off q_c. Values of the exponent lower than q_c are set to q_c with the shear then set to zero.

$a =$ qprof_coef(1)	1/length for exponential increase
$q_v =$ qprof_coef(2)	Multiplication factor
$q_c =$ qprof_coef(2)	Cut off value

$$
\begin{aligned}
q &= \max(q_v \psi \exp(a\psi), q_c) \\
\hat{s} &= 1 + \psi a \quad \text{for } q > q_c
\end{aligned}
$$

7.5.14 q-profile option: mishchenko

This has for example been used in Mishchenko and Zocco, Phys. Plasmas 19.

$q_0 = \mathrm{qprof\_coef}(1)$	The q-value on the axis
$a = \mathrm{qprof\_coef}(2)$	Norm factor for ψ
$\nu = \mathrm{qprof\_coef}(2)$	Exponent

$$q = q_0 + (1 - q_0)(\psi/a)^\nu$$

$$\hat{s} = \frac{\nu(1 - q_0)(\psi/a)^\nu}{q}$$

Chapter 8

Numerical implementation

In this section we outline the details of the numerical solution of the equations. GKW uses a combination of finite difference and spectral methods. The turbulence in the plane perpendicular to the magnetic field is homogeneous and the solution in the ζ, ψ plane is represented by Fourier modes, as mentioned previously. All other directions are treated using finite difference techniques. In what follows $N_{\rm mod}$ and N_x are the number of bi-normal (ζ direction) and radial (ψ direction) modes respectively, N_{sp} is the number of kinetic species, N_s the number of grid points along the field line. $N_{v\|}$ is the number of grid points in the parallel velocity direction and N_μ the number of magnetic moment grid points.

8.1 Derivatives along the magnetic field line and in the parallel velocity direction

Terms I, IV and VII on the right hand side of Eq. (7.37), involving a derivative along the magnetic field line or in the parallel velocity direction are considered here. These terms correspond to an advection with a spatially dependent advective velocity and can be written in the form $-v(s)\frac{\partial g}{\partial s}$. The second and fourth-order centred-differences used for these terms can then be written

$$v\frac{\partial g}{\partial s} \rightarrow v_i \frac{-g_{i-1} + g_{i+1}}{2\Delta s} - D|v_i|\frac{g_{i-1} - 2g_i + g_{i+1}}{2\Delta s}, \tag{8.1}$$

$$v\frac{\partial g}{\partial s} \rightarrow v_i \frac{g_{i-2} - 8g_{i-1} + 8g_{i+1} - g_{i+2}}{12\Delta s} - D|v_i|\frac{-g_{i-2} + 4g_{i-1} - 6g_i + 4g_{i+1} - g_{i+2}}{12\Delta s}, \tag{8.2}$$

where the terms involving a dissipation coefficient D correspond to (hyper)diffusive upwind dissipation. This dissipation is not included in the case of term VII (containing derivatives of ϕ) for numerical stability reasons.

Merely to keep the code simple, the finite difference expression for the 4th derivative in Eq. 8.2 has only second order accuracy, as a higher order would involve more than just 5 gridpoints.

The fact that in Fourier representation the ∂_s^4 becomes ik_s^4 illustrates that the dissipation term affects in particular the small scales.
Note that only a small dissipative contribution is desired, the smaller the higher the grid resolution is. Hence it makes sense to prepend the dissipation term with a factor Δs^n, where n is an arbitrary power. This should make an appropriate choice of the coefficient D (parameters *disp_par* and *disp_vp*) more obvious.

$$-D\Delta s^n|v_i|\frac{-g_{i-2} + 4g_{i-1} - 6g_i + 4g_{i+1} - g_{i+2}}{12\Delta s^4}$$

One may consider a grid-scale oscillation, i.e. $g_i = 1$ for even indices i and -1 for odd i. This is just the kind of oscillation that the artifical dissipation term is supposed to suppress. In this example, the finite difference in the dissipation term yields $1/\Delta s^4$ while that of the advection yields $1/\Delta s$. If one prepends the dissipation with Δs^3 then for coefficients $D \sim 1$ the dissipation term is comparable to the advection term and will indeed damp the grid-scale oscillation effectively. This is why in Eq. 8.2 only Δs appears, not Δs^4.

Rather than apply this central differencing scheme separately to terms I and IV, the code user can choose an alternative scheme, which combines the derivative along the field line with the trapping term. Defining $H(s, v_\parallel) = \frac{1}{2}v_\parallel^2 + \mu B + \frac{1}{2}\mathcal{E}_R$, and noting that $B\mathcal{G} = \mathcal{F}\frac{\partial B}{\partial s}$ in IV, we can combine terms I and IV to give

$$v_R \mathcal{F}\left(\mu \frac{\partial B}{\partial s}\frac{\partial \hat{f}}{\partial v_\parallel} - v_\parallel \frac{\partial \hat{f}}{\partial s}\right) = v_R \mathcal{F}\ \{H, \hat{f}\}, \tag{8.3}$$

where $\{H, \hat{f}\}$ is a Poisson bracket (note that we have neglected the redundant dependence of μ in the definition of H for simplicity). Following the second and fourth-order schemes of Arakawa [76], we allow the Poisson bracket to be differenced directly. The terms containing D in Eqs (8.1) and (8.2) are also added to allow some dissipation. The advantage of this scheme over the separate differencing scheme is that usually zero (or relatively small) dissipation is needed to obtain a stable solution. Presently this is implemented only as a fourth-order scheme.

8.2 Boundary conditions along a field line

We here discuss the implementation of the boundary conditions in the parallel direction s, that are presented in Sec. 2.3. After one poloidal turn the Fourier mode connects to a mode with a different radial wave vector. The grid in the s direction is therefore constructed such that at the end of the field line it connects smoothly to the start of the field line (i.e. the grid spacing remains constant). Of course, the radial wave vector to which the mode has to connect might not be represented on the grid. At the end (or beginning) of the field line a boundary condition is then employed. GKW allows for a choice in this boundary condition, setting the perturbed distribution either to zero (NO LONGER AVAILABLE) or to have a zero derivative (default, *parallel_boundary_conditions='open'*) in the direction upwind with respect to the convection.[1] The down wind direction is always treated with a zero derivative boundary condition, allowing the distribution to 'flow off the grid', without reflections. Close to the end point boundaries a staggered change in differencing order accuracy is used to remove ghost cells and to allow the flow of the distribution function out of the domain. Reflection or the allowed build up at the upper velocity boundaries is seen as unphysical and therefore undesirable. Because of the use of up-winding the boundary condition is dependent on the sign of the advective velocity.

When the advective velocity $v(s)$ is positive the distribution function will move toward the left boundary, and the following scheme is used

$$\begin{aligned}
&\text{Boundary cell :} \qquad \text{Second order scheme} - \frac{\partial g}{\partial s} \rightarrow \frac{-\frac{3}{2}g_n + 2g_{n+1} - \frac{1}{2}g_{n+2}}{\Delta s}, \\
&\text{Adjacent cell :} \qquad \text{Third order} - \frac{\partial g}{\partial s} \rightarrow \frac{-\frac{1}{3}g_{n-1} - \frac{1}{2}g_n + g_{n+1} - \frac{1}{6}g_{n+2}}{\Delta s}.
\end{aligned}$$

with the right hand boundary using the centrally differenced schemes. When the advective velocity is negative the opposite is true and the right hand boundary is differenced in an identical way. To obtain the scheme for the opposing boundary simply reverse the signs of the equation above. When using the second order scheme, special consideration is needed only for the boundary cell. Depending on the direction of the advective velocity, a first order back-winded scheme is used at the boundary. The effect of these open boundary conditions is to remove the need for ghost cells in the differencing scheme while maintaining a high order scheme. Compared to other alternatives we have found that the chosen scheme removes rapid fluctuations near the end points, and therefore allows for a minimum dissipation. In the Poisson bracket formulation Eq. 8.3, the above convention is integrated into the scheme at the boundaries to have similar properties. Less (or no) dissipation (Diffusion/super-diffusion) is needed for this scheme, but the dissaption

[1]The option *parallel_boundary_conditions='Dirichlet'* may be different to both of these and is experimental - not documented further here.

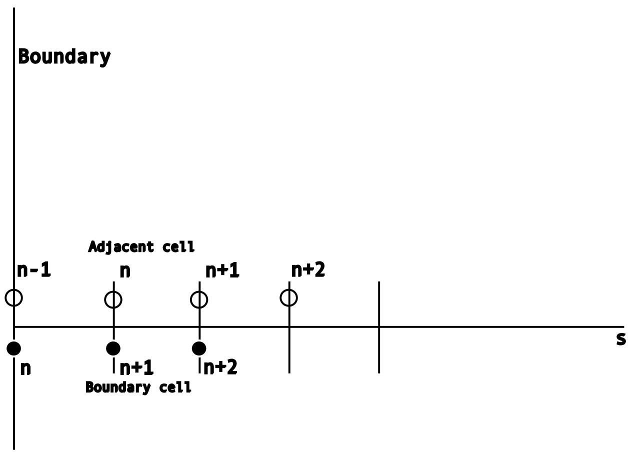


Figure 8.1: Schematic showing the cells of intest in the differencing scheme when considering the boundary cell (Hollow circles) and the adjacent cell (Filled circles). The figure shows the left hand boundary, for right hand simply reflect in the y direction

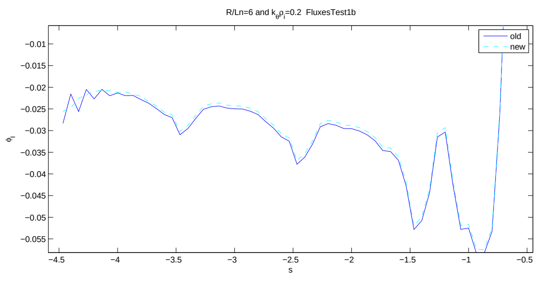


Figure 8.2: Figure showing the effect of the new open boundary conditions at the boundary. The new boundary has reduced cell to cell fluctuations giving a smoother solution.

is implemented as seperate routine using the normal differencing above, allowing identical dissipation to be applied to both schemes.

The effect of the open boundary conditions is to remove the need for ghost cells in the differencing scheme while maintaining a high order scheme. It has reduced the amplitude of cell to cell fluctuations refelecting off the boundary (See figure 8.2), which in turn, has reduced the need for adding artificial dissipation to the system to damp away these modes. The dissipation, however, can not be wholly turned off in non-linear runs. Note also that the dealiasing in nonlinear runs also has some dissipative effect in the perpendicular spatial directions.

8.3 Parallel velocity grid - Trapping condition

(STILL EXPERIMENTAL AND UNDER DEVELOPMENT) By contorting the velocity grid to conform to the electron trapping condition we are able to remove terms that involve derivatives in the parallel velocity direction. This greatly speeds up linear run computation times but more significantly removes the need for numerical diffusion in the parallel velocity direction and lastly, provides resolution in the required areas.

This option is turned on by setting *vp_trap* = 1 in the *CONTROL* namelist of the input deck. A further parameter is needed in the *GRIDSIZE* namelist which is named *n_trapped* which is the number of grid points within the trapped region of the velocity grid.

In the current implementation of the trapping condition, the number of cells in the s direction and the number of points in the parallel velocity direction add certain restrictions to the number of points that can be placed within the trapped region.

Firstly, for ease of differencing the bounce points of a specific $v_{\|0}$ (The velocity on the low-field side) must occur exactly half way between two grid-cells, thus the number of grid cells in the s direction restrict both the number and initial values of $v_{\|}$. The maximum number of bounce points available is given by $n\_s\_grid/2-1$. The value of *n_trapped* must be less than or equal to this. If less is chosen then the code will select which points to use determined by the spacing between points (The closer to the boundary, the closer the bounce points become), the algorithm will iterate over the points, sequentially removing points until *n_trapped* are left. *n_trapped* must also be less than *n_vpar_grid*. Outside the trapping region the points are uniformly placed, the spacing is determined by the number *n_vpar_grid* $-2\times$ *n_trapped*.

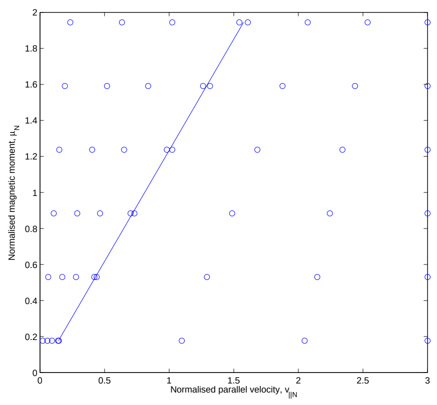


Figure 8.3: Figure of parallel velocity grid points as given by the trapping condition, the line corresponds to the trapped/untrapped boundary $v_{\|0} = \sqrt{2\mu(B_{max} - B_{min}) + \mathcal{E}^H_\Omega - \mathcal{E}^L_\Omega}$. In the above data, N_vpar_grid = 16 (only one half of the grid is shown as it is symmetric),n_trapped = 4 and $\mu = 6$. It must be noted that the total number of points within the trapped region (full grid) is 2×n_trapped.

This algorithm gives the grid at the low field side (For an example see figure 8.3), this is then propagated around the torus by the equation

$$v_{\|N} = \sqrt{v^2_{\|0N} + 2\mu_N(B - B_0) + \mathcal{E}_\Omega - \mathcal{E}^L_\Omega} \tag{8.4}$$

where B_0 is the magnetic field at the low field side, and $\mathcal{E}^L_\Omega$ is the centrifugal energy at the low field side. When a bounce point is reached, the velocity becomes zero, and all velocity grid points after this along s are set to zero and are ignored in the differencing. The grid is also symmetric around $v_{\|} = 0$.

Rules that must be followed, otherwise the code will throw you out,

- The value of *n_trapped* must be less than *n_vpar_grid*/2
- *n_trapped* must also be less than or equal to *n_s_grid*/2−1
- For optimal convergence 2×*n_trapped* must be roughly half of *n_vpar_grid*. (This may no longer be true: See issue 8 for latest updates on Gaussian integration (and the separate document Gaussian_integration.tex which should be merged here)

8.4 The collision operator

The collision operator can be thought of as a convection/diffusion equation in velocity space and thus is differenced accordingly. The velocity space boundary conditions are taken to be zero flux, which occurs naturally at the $\mu = 0$ boundary and is artificially imposed on the other three boundaries. These boundary conditions guarantee the conservation of mass. Considering that the diffusion coefficient is velocity space dependent, and the grid can be non-uniform in the μ-direction, the differencing scheme of the equations is chosen as follows for the parallel velocity direction

$$\frac{\partial}{\partial v_\parallel}\left(D\frac{\partial \hat{f}}{\partial v_\parallel}\right) \to \frac{1}{v^j_{i+1/2\parallel} - v^j_{i-1/2\parallel}}\left(D^j_{i+1/2}\left(\frac{f^j_{i+1} - f^j_i}{v^j_{i+1\parallel} - v^j_{i\parallel}}\right) - D^j_{i-1/2}\left(\frac{f^j_i - f^j_{i-1}}{v^j_{i\parallel} - v^j_{i-1\parallel}}\right)\right), \tag{8.5}$$

where i denotes the point in the parallel velocity direction and j the μ direction. When considering the cross-terms a four point interpolation method is used to calculate the half point values. Four points are needed on a two-dimensional grid. The friction term is differenced in the same way, but with a two point interpolation.

To bypass the complications caused by non-uniform distribution of μ points (which in uniform in the $v_\perp$ coordinate), the code chooses the suitable version of the collision operator that is uniform, and as such, can conserve mass (and parallel momentum) to machine accuracy.

A further switch **selfcollcon** in the collisions input deck changes the momentum conservation from just the self-collisions for each species (selfcollcon = .true.) to absolute conservation (selfcollcon=.false, default=.true.). Note that the absolute conservation option is unphysical and only implemented for tests and benchmarks purposes.

8.5 The Fourier representation

In a nonlinear run a rectangular grid of wave vectors (k_ζ, k_ψ) is used. The nonlinearity is treated using fast Fourier transforms (FFT). Different choices of FFT libraries are possible, but at present we use mostly the FFTW library [56]. For the nonlinear term a de-aliasing method is applied, i.e. the transformation to real space is constructed on a grid finer (number of grid points at least 3/2 larger in real space) than the Fourier modes represent. The de-aliasing method is non-conservative and has a dissipative effect on the finer grid scales, with the consequence that a numerical dissipation term in the spectral directions is generally not required.

Since the distribution function is a real quantity the number of complex Fourier modes can be reduced by a factor two and a half plane of wavevectors with $k_\zeta \geq 0$ is used. Grid sizes in real space are chosen so that they have only small prime factors. Such sizes lead to particularly efficient computations with the standard FFT libraries. There are no significantly more efficient grid sizes any more, as it used to be in the past.

8.6 Time integration

GKW has several options for the explicit time integration: modified midpoint, fourth order Runga Kutta (recommended), and a second order scheme that is stable for hyperbolic equations. An implicit scheme is also implemented, but is still under development and not yet recommended for general use (For latest updates see issue 10).

8.7 Poisson equation

The Poisson equation 2.277 is split into three linear terms which are stored in the main solution matrix, but evaluated separately from the other linear terms. The calculation of the fields takes place in `fields.F90` subroutine *calculate_fields*.

The Poisson equation given in 2.277 can be rewritten.

$$\sum_{sp,ions} Z_{sp} n_{R_0,sp}\left[2\pi B\int \mathrm{d}v_\parallel \mathrm{d}\mu J_0(k_\perp\rho_{sp})\hat{g}_{sp} + \frac{Z_{sp}}{T_{Rsp}}[\Gamma(b_{sp})-1]\exp(-\mathcal{E}_{R\mathrm{sp}})\hat{\phi}\right] = \frac{n_{R_0,e}\exp(-\mathcal{E}_{Re})}{T_{Re}}(\hat{\phi}-\left\{\hat{\phi}\right\}_{\mathsf{FSA}})$$
$$\sum_{sp,ions} Z_{sp} n_{R_0,sp}\left[2\pi B\int \mathrm{d}v_\parallel \mathrm{d}\mu J_0(k_\perp\rho_{sp})\hat{g}_{sp}\right] + \frac{n_{R_0,e}\exp(-\mathcal{E}_{\mathrm{Re}})}{T_{Re}}\left\{\hat{\phi}\right\}_{\mathsf{FSA}} =$$
$$= -\sum_{sp,ions} Z_{sp} n_{R_0,sp}\left[\frac{Z_{sp}}{T_{Rsp}}[\Gamma(b_{sp})-1]\exp(-\mathcal{E}_{R\mathrm{sp}})\hat{\phi}\right] + \frac{n_{R_0,e}\exp(-\mathcal{E}_{\mathrm{Re}})}{T_{Re}}\hat{\phi} \tag{8.6}$$

Solving for $\hat{\phi}$

$$\sum_{sp,ions} Z_{sp} n_{R_0,sp}\left[2\pi B\int \mathrm{d}v_\parallel \mathrm{d}\mu J_0(k_\perp\rho_{sp})\hat{g}_{sp}\right] + \frac{n_{R_0,e}\exp(-\mathcal{E}_{\mathrm{Re}})}{T_{Re}}\left\{\hat{\phi}\right\}_{\mathsf{FSA}} =$$
$$= \left[-\sum_{sp,ions} Z_{sp} n_{R_0,sp}\left[\frac{Z_{sp}}{T_{Rsp}}[\Gamma(b_{sp})-1]\exp(-\mathcal{E}_{R\mathrm{sp}})\right] + \frac{n_{R_0,e}\exp(-\mathcal{E}_{\mathrm{Re}})}{T_{Re}}\right]\hat{\phi} \tag{8.7}$$

yields

$$\hat{\phi} = -\frac{1}{A}\left[\sum_{sp,ions}\underbrace{Z_{sp}n_{R_0,sp}2\pi B\int \mathrm{d}v_\parallel \mathrm{d}\mu J_0(k_\perp\rho_{sp})\,\hat{g}_{sp}}_{\textit{poisson\_int()}\to M^3} + \underbrace{\frac{n_{Re}\exp(-\mathcal{E}_{Re})}{T_{Re}}\left\{\hat{\phi}\right\}_{\mathsf{FSA}}}_{\textit{poisson\_zf()}\to\text{eq. } 8.15}\right] \tag{8.8}$$

where the factor

$$A = \sum_{\mathrm{sp,ions}} n_{sp}Z_{sp}\left[\frac{Z_{sp}}{T_{Rsp}}(\Gamma(b_{sp})-1)\exp(-\mathcal{E}_{Rsp}) - \frac{\exp(-\mathcal{E}_{\mathrm{Re}})}{T_{Re}}\right] = M^4 \tag{8.9}$$

is evaluated in subroutine *poisson_dia*. The symbols M^3 and M^4 number the matrices presented in this paragraph (and are not "powers of M" or so) and refer to sections of *mat*. In this sense also M^1 and M^2 which contain the linear terms of the GKE and are mentioned elsewhere denote sections 1 and 2 of the matrix *mat*. The quantity $\left\{\hat{\phi}\right\}_{\mathsf{FSA}}$ can be eliminated from 8.8 by taking the flux surface average of equation 8.8.

$$\left\{\hat{\phi}\right\}_{\mathsf{FSA}} = \left\{-\frac{1}{A}\left[\sum_{sp,ions} Z_{sp}n_{R_0,sp}2\pi B\int \mathrm{d}v_\parallel \mathrm{d}\mu J_0(k_\perp\rho_{sp})\hat{g}_{sp} + \frac{n_{Re}\exp(-\mathcal{E}_{\mathrm{Re}})}{T_e}\left\{\hat{\phi}\right\}_{\mathsf{FSA}}\right]\right\}_{\mathsf{FSA}}$$
$$= \left\{-\frac{1}{A}\sum_{sp,ions} Z_{sp}n_{R_0,sp}2\pi B\int \mathrm{d}v_\parallel \mathrm{d}\mu J_0(k_\perp\rho_{sp})\hat{g}_{sp}\right\}_{\mathsf{FSA}} + \left\{-\frac{1}{A}\frac{n_{Re}\exp(-\mathcal{E}_{\mathrm{Re}})}{T_{Re}}\left\{\hat{\phi}\right\}_{\mathsf{FSA}}\right\}_{\mathsf{FSA}} \tag{8.10}$$

The expressions containing $\left\{\hat{\phi}\right\}_{\mathsf{FSA}}$ are drawn together:

$$\left\{\hat{\phi}\right\}_{\mathsf{FSA}} - \frac{n_{Re}}{T_{Re}}\left\{\frac{-\exp(-\mathcal{E}_{\mathrm{Re}})}{A}\right\}_{\mathsf{FSA}}\left\{\hat{\phi}\right\}_{\mathsf{FSA}} = \left\{-\frac{1}{A}\sum_{sp,ions} Z_{sp}n_{R_0,sp}2\pi B\int \mathrm{d}v_\parallel \mathrm{d}\mu J_0(k_\perp\rho_{sp})\hat{g}_{sp}\right\}_{\mathsf{FSA}}$$

$$\left\{\hat{\phi}\right\}_{\mathsf{FSA}}\left(1 - \frac{n_{Re}}{T_{Re}}\left\{\frac{-\exp(-\mathcal{E}_{\mathrm{Re}})}{A}\right\}_{\mathsf{FSA}}\right) =$$
$$\frac{n_{Re}}{T_{Re}}\exp(-\mathcal{E}_{\mathrm{Re}})\left\{\hat{\phi}\right\}_{\mathsf{FSA}}\left(\frac{T_{Re}}{n_{Re}}\frac{1}{\exp(-\mathcal{E}_{\mathrm{Re}})} - \left\{\frac{-1}{A}\right\}_{\mathsf{FSA}}\right) = \tag{8.11}$$

The zonal correction term can then be written as a function of the distribution function, parameters and background quantities.

$$\frac{n_{Re}\exp(-\mathcal{E}_{\rm Re})}{T_{Re}}\left\{\hat{\phi}\right\}_{\sf FSA} = \frac{\left\{-\frac{1}{A}\sum_{sp,ions} Z_{sp}n_{R_0,sp}2\pi B\int {\rm d}v_\parallel {\rm d}\mu J_0(k_\perp\rho_{sp})\hat{g}_{sp}\right\}_{\sf FSA}}{\frac{1}{\exp(-\mathcal{E}_{\rm Re})}\left(\frac{T_Re}{n_Re}+\left\{\frac{\exp(-\mathcal{E}_{\rm Re})}{A}\right\}_{\sf FSA}\right)} \tag{8.12}$$

The subroutine *poisson_zf* prepares matrices according to the definitions

$$Z = -\frac{{\rm d}s}{A} \tag{8.13}$$

$$Y = \frac{1}{\exp(-\mathcal{E}_{\rm Re})}\left(\frac{T_Re}{n_Re}+\left\{\frac{\exp(-\mathcal{E}_{\rm Re})}{A}\right\}_{\sf FSA}\right) \tag{8.14}$$

which are used to express the right hand side of 8.12 in the form

$$\frac{n_{Re}\exp(-\mathcal{E}_{\rm Re})}{T_{Re}}\left\{\hat{\phi}\right\}_{\sf FSA} = \frac{\sum_s ZM^3\hat{g}}{Y}\ . \tag{8.15}$$

Thanks to 8.12, 8.8 allows the subroutine *calculate_fields* to calculate $\hat{\phi}$ explicitely.

To calculate $\hat{\phi}$ from the distribution function, first the integral part *poisson_int* is evaluated using section M^3 of the matrix.

$$a_i(k_y,k_x,s) = M^3_{ij}\hat{g}_j. \tag{8.16}$$

Next, the zonal flow correction term, equation 8.12, is evaluated separately, using the special "matrices" Z and Y. First, the flux surface average in the numerator is evaluated.

$$b_i(k_x) = \sum_s Z_{ij}a_j \tag{8.17}$$

Then each element is divided by the denominator from equation 8.12.

$$c_i(0,k_x,s)) = b_i/Y_i(s) \tag{8.18}$$

The zonal correction term is then subtracted from the minuend in equation 8.8.

$$d_i(k_y,,k_x,s) = a_i - c_i \tag{8.19}$$

Finally, all elements are divided by the factor A (A being identical to M^4, the section 4 of "the matrix of the linear terms").

$$\hat{\phi}_i(k_y,k_x,s) = d_i/M^4_i \tag{8.20}$$

The other linear terms are stored in sections 1 and 2 of the matrix and evaluated as $\hat{f}_i{}' = \hat{f}_i + \delta t\cdot\hat{f}_j M^{1,2}_{ij}$ in *calculate_rhs* after the evaluation of the fields.

8.8 Notes on individual routines

This section discusses the individual routines and gives some details on the physics processes implemented. Here, not all the input and output or all the variables are discussed. It merely gives some details that should allow the reader to understand the code better.

8.8.1 linear_terms.f90

NOT INCLUDED here: centrifugal effects

In the treatment so far, the terms as implemented in the code were described in 7.37 as terms I-VIII. Here we link these terms to the routines in the code, as well as to 1.31, the Vlasov equation for the modified perturbation distribution function g.

$$\frac{\partial g}{\partial t} + (\underbrace{v_\parallel \mathbf{b}}_{I} + \underbrace{\mathbf{v}_D}_{II}) \cdot \nabla f + \underbrace{\mathbf{v}_\chi}_{III} \cdot \nabla g - \underbrace{\frac{\mu B}{m} \frac{\mathbf{B} \cdot \nabla B}{B^2} \frac{\partial f}{\partial v_\parallel}}_{IV} = S. \tag{8.21}$$

Where the source from the Maxwellian background is given by eqn 1.32

$$S = -(\underbrace{\mathbf{v}_\chi}_{V} + \underbrace{\mathbf{v}_D}_{VI}) \cdot \nabla_p F_M - \frac{Ze}{T} [\underbrace{v_\parallel \mathbf{b}}_{VII} + \underbrace{\mathbf{v}_D}_{VIII}] \cdot \nabla \chi F_M \tag{8.22}$$

We also need the expansion of the drift velocity 1.21

$$\begin{aligned} \mathbf{v}_D &= \underbrace{\frac{1}{Ze} \left[\underbrace{\frac{m v_\parallel^2}{B}}_{1} + \underbrace{\mu}_{2} \right] \frac{\mathbf{B} \times \nabla B}{B^2}}_{A} + \underbrace{\frac{m v_\parallel^2}{2ZeB} \beta' \mathbf{b} \times \nabla \psi}_{3} + \underbrace{\frac{2 m v_\parallel}{ZeB} \mathbf{\Omega}_\perp}_{4} \\ &- \underbrace{\frac{m \Omega^2 R}{ZeB} \mathbf{b} \times \nabla R,}_{5} \end{aligned} \tag{8.23}$$

In linear_terms.f90 the various terms from these equations are input into the matrix in individual routines which will be described below. Most of the computation in the code is done in the wave vector domain. All derivatives in the perpendicular direction are in Fourier space; but derivatives along the field line are done in real space (terms I and VII)

Some routines in linear_terms.f90 have two versions (identified by a postfix to the routine name), one for each of the second order and fourth order methods. We concentrate here on the fourth order method since the other is out of order. The routines do the same thing with regards to which terms are implemented; the implementation is different for each method. The linear terms all operate on the the distribution g (often called f in the code comments!), because the linear terms matrix is applied to **the array called *fdisi* which always contains g.**

calc_linear_terms is the master routine that calls the subroutines which put the terms I-XI (excluding term III) into the matrix. Broadly, the first section puts the terms on the LHS of 8.21. The second section calculates the source terms due to the maxwell background.8.22. The third section of *calc_linear_terms* calls the routines that input the field equations into the matrix. These are the equations that force the solution the be self consistent with the potentials calculated from the distribution function. After each section is completed there is a call to *finish_matrix_section* in module *matdat* which stores the k index labels of the matrix for each section..

Hence *calc_linear_terms* covers all the terms I-VII in 7.37 (also labelled as the underbraced numbers in 8.21, 8.22, and 8.23), with the exception of term III, which is implemented in non_linear_terms.f90.

vpar_grad_df, term I

vpar_grad_df calculates the convection parallel to the field. This routine also adds parallel velocity dissipation, which can be required for a numerically stable solution in nonlinear runs. The dissipation is controlled by the input parameter *disp_par*. This term is the free streaming motion along the field line, with derivative calculated in real space.

vdgradf, term II

vdgradf calculates the drift in the gradient of the eikonal term.

trapdf, terms IV

trapdf calcuates the mirror trapping terms dependent on the switch variable *trapping*. This routine also adds perpendicular velocity dissipation, which can be required for a numerically stable solution in nonlinear runs. The dissipation is controlled by the input parameter *disp_vp*.

ve_grad_fm, term V

ve_grad_fm calculates the $\mathbf{E} \times \mathbf{B}$ drift due to the background distribution, as well as (optionally according to logical *nlapar*) the electromagnetic correction $\mathbf{v}_{\delta B_\perp}$.

neoclassical, term VI

neoclassical optionally includes the neoclassical terms according to the switch *neoclassics*. Note many gyrokinetics codes disregard this term.

vpgrphi, term VII

vpgrphi calculates the landau damping term, dependent on the switch variable *landau*. This is an acceleration along the field line with parallel derivative in real space.

vd_grad_phi_fm, term VIII

vd_grad_phi_fm calculates the drift in the gradient of phi times the velocity derivative of the Maxwell background. Note that Equation terms 8.23.1 and 8.23.2 are combined as 8.23.A in the E_D.multiplier as defined in 7.37.

8.8.2 non_linear_terms.F90

CHECK CONSISTENCY OF THIS SECTION

The Fourier amplitudes of the potential (and perturbed distribution function) are defined as

$$\phi_N(x) = \sum_k \phi_{Nk} \exp[\mathrm{i}kx] + c.c. \tag{8.24}$$

Note that because the inverse discrete Fourier transform (H_l) is defined such that the values in real space (h_n) are given by

$$h_n = \frac{1}{N} \sum_{l=0}^{N-1} H_l \exp[\mathrm{i}2\pi l n/N] \tag{8.25}$$

where N is the total number of grid points of the discrete transform. The relation between the amplitudes and the discrete Fourier transform, therefore, is

$$H_l = N\phi_{Nk} \tag{8.26}$$

The wave vector k is given by the standard relation

$$k = \frac{2\pi l}{N\Delta} \tag{8.27}$$

where Δ is the grid distance. This relation is used to determine the size of the box in real space. It follows that the lowest non-zero wave vector k_1 determines Δ and the size of the box $L = N\Delta$

$$L = \frac{2\pi}{k_1} \tag{8.28}$$

***add_non_linear_terms*, term III**

This routine implements the full term III with electromagnetic corrections. Hence the routine finds the $\mathbf{E} \times \mathbf{B}$ drift of the perturbed distribution function, and adds the term:

$$\mathbf{v}_\chi \cdot \nabla g = \frac{\mathbf{b} \times \nabla \langle \hat{\chi} \rangle}{B} \cdot \nabla g \tag{8.29}$$

Without the elctromagnetic corrections, Term III of 7.37 becomes

$$\text{III} = -\mathbf{v}_\phi \cdot \nabla g \mathcal{T}\Big(\mathcal{E}^{\psi\zeta}\Big[\mathcal{T}^{-1}(ik_\zeta\hat{\phi})\mathcal{T}^{-1}(ik_\psi\hat{g}) - \mathcal{T}^{-1}(ik_\zeta\hat{g})\mathcal{T}^{-1}(ik_\psi\hat{\phi})\Big]\Big), \tag{8.30}$$

where the expansion of the tensor sum uses the anti-symmetry of $\mathcal{E}^{\beta\alpha}$. Note also that the diagonal elements of $\mathcal{E}^{\beta\alpha}$ are zero, and derivatives along s are neglected.

The gyroaveraged potential $\langle\phi_N\rangle$ is first calculated in the wavevector domain by multiplying the potential by the Bessel function J_0 (2.259). In k space the derivative of the potential is obtained by multiplying by ik_α.

Hence the implementation in the code is

$$a = iJ_0(k_\perp\rho)k_{N,\zeta}\phi_{N,k} = ik_\zeta\rho_{\text{ref}}\langle\phi_{N,k}\rangle \tag{8.31}$$

$$b = iJ_0(k_\perp\rho)k_{N,\psi}\phi_{N,k} = ik_\psi\rho_{\text{ref}}\langle\phi_{N,k}\rangle \tag{8.32}$$

Note the ρ in the Bessel function is species dependent. Also the normalization of the k vector $k = k_N/\rho_{\text{ref}}$ introduces a multiplier of ρ_{ref} Hence the inverse Fourier transformation gives

$$ar = \mathcal{F}^{-1}(a) = \rho_{\text{ref}}\frac{\partial\langle\phi_N\rangle}{\partial\zeta_N} = \frac{\rho_{\text{ref}}}{R_{\text{ref}}}\frac{\partial\langle\phi_N\rangle}{\partial\zeta} = \rho_*\frac{\partial\langle\phi_N\rangle}{\partial\zeta} \tag{8.33}$$

$$br = \mathcal{F}^{-1}(b) = \rho_*\frac{\partial\langle\phi_N\rangle}{\partial\psi} \tag{8.34}$$

Since the coordinates are normalised by $x_{\alpha,N} = x_\alpha/R_{\text{ref}}$. Here we have explicitly linked the normalizations of the k vector and coordinates to show that under the Fourier transform; $\rho_*\frac{\partial}{\partial x_\alpha} \to \text{i}k_\alpha$, as was intended in its definition. Put simply, a factor of ρ_* appears after the inverse Fourier transform.

Similarly, the gradient of the distribution function in k space is found $a = ik_\psi g_{N,k}$, $b = ik_\zeta g_{N,k}$ and applying the inverse Fourier transform gives

$$cr = \mathcal{F}^{-1}(a) = \rho_*\frac{\partial g_N}{\partial\psi} \tag{8.35}$$

$$dr = \mathcal{F}^{-1}(b) = \rho_*\frac{\partial g_N}{\partial\zeta} \tag{8.36}$$

The tensor $\mathcal{E}^{\alpha\beta}$ (see 2.23) is defined in geom.f90 with name $e$fun. Note that although the tensor has two indices, each component varies along the field line due to the gradient in B. The array therfore has 3 indices with $efun(i, 2, 1)$ being the $\mathcal{E}(s)^{\zeta\psi}$ component. Calling this element of the array we now have all the components needed to calculate 2.263. The factor of ρ_*^2 comes from the 2 inverse Fourier transforms. Term III then appears in the code as

$$efun(i, 2, 1) * (ar * cr - br * dr) \tag{8.37}$$

Finally this is Fourier transformed back to the wavevector domain and $\mathcal{F}(\mathbf{v}_E \cdot \nabla g)$ is added to the RHS of the equations - which were passed to the subroutine as an argument along with the distribution function when it was called from exp_integration.F90.

The routine also calculates maximum velocities to be used in the timestep estimator. The first time this routine is called it does some initialisations which are not repeated in later calls: The sizes of the boxes in real space are calculated, the max k-vectors are calculated and the location of the $k_x = 0$ mode is determined. Index arrays are also set up to relate the storage regime on the fast Fourier transform (fft) routine to the GKW storage regime. The remainder of the routine as described above is executed every time.

8.9 The code

The package GKW consists of a README file (where a description of the full contents can be found), Fortran 95 source files, various makefiles and related scripts, some documentation, some sample code input files, some scripts associated with running and testing the code and tools to pre/post process data and perform visualisation. This section describes what the code can do in terms of parallelisation and how it is structured with respect to the solution of the equations presented in the previous sections. In order to do this, we follow the various tasks the code performs as it runs and discuss the relevant modules to that task. Instructions for building, installing and running the code can be found in section 9.

Each source code file contains a module, except the main program file linart.f90. Each source file has either the extension `.f90` or `.F90`, the latter denoting the need to be pre-processed. For the modules, each module name corresponds directly to the file name (which is always lower case), minus the suffix. Therefore, in the following subsections, when referring to the module *GRID*, the corresponding source code can be found in the file grid.F90.

The code does not come with any routines to perform FFTs, which are required for the computation of the non-linear term in Eq. 2.263, and to transform some of the diagnostics into real space. No FFT's are required for a linear run and the code can be compiled without FFT functionality. The FFT's are presently performed via the external FFTW3 library [56], for which the *FFT* module provides an interface. Although more interfaces for various other portable Fortran FFT libraries could be provided, the performance of the FFTW3 library has been found to exceed that of other alternatives significantly. In addition, the FFTW3 library is fairly portable, so to date there has been no good reason to consider alternatives.

GKW is parallelised primarily using the Message Passing Interface (MPI) [2]. Each parallel process is responsible for solving the equations over a sub domain of the space and a subset of the species. A process usually only has knowledge of the part of the solution it is responsible for, unless explicitly communicated between processes using the MPI library. The parallelisation is discussed in more detail in a later section.

8.9.1 Initialisation

At the start of a run, the code will first call various basic MPI routines. These give each process an individual rank (*processor_number*) and inform each process of the total number of processors available for the computation (*number_of_processors*). Checks are then performed to see if various output or restart files are present.

After this, the main input file is read. A single processor is responsible for doing this. The input file contains several Fortran namelists, which are described later. The first namelist is called *control* and is read in the

[2]Shared memory parallelisation, implemented via OpenMP directives, is now possible with the code and is documented in a later subsection

module *CONTROL*, and is responsible for controlling the basic switches for the physics, numerics and some of the code output. Each module requiring input is responsible for reading its input from this file. After each namelist is read, the processor that read the file broadcasts the information to all the other processors via MPI. All processors can then individually check that the input is allowed or satisfactory, and if necessary, abort the run.

8.9.2 Parallelisation: local grid

The sizes of the local and global grids and associated quantities are dealt with in the module *GRID*. The namelist *gridsize* contains the input variables for the grid sizes. The code can decompose the calculation over the s, $v_\parallel$ and μ coordinates and the species. Given the number of processors as input, the routines of *GRID* decide how the global solution will be subdivided between the processors. Alternatively, a user can specify explicitly the number of processors over which to subdivide the problem in each divisible direction. If a user explicitly sets an invalid combination, the code will abort at this point or earlier. For the decomposition over the species or the μ grid, 1 point per processor is the minimum. The minimum number of grid points per processor in the s and $v_\parallel$ directions is limited to either 1 (when using the second-order finite differences) or 2 (when using the fourth-order scheme). These minimum grid sizes dictate the maximum number of processors that can be used in a given direction; the maximum is either the number of grid points in that direction (for the second-order scheme) or half that value (for the fourth-order scheme). The parallelisation is discussed in more detail in Sect. 8.10.

A processor is provided with various logical variables denoting if it is at a particular boundary in the global domain together with the ranks of the processors that it will need to communicate with (which are usually the ones responsible for adjacent parts of the computational domain).

Once the local grid sizes have been confirmed, some MPI communicators are set up to facilitate communication between processes responsible various subsets of the whole domain. These are found in the *MPICOMMS* module. They are particularly useful when performing a sum over a subset of the coordinates when all the relevant points are not present on the local processor.

8.9.3 Memory layout and allocation

In GKW, the solution pertubation distribution $g(k_\zeta, k_\psi, s, \mu, v_\parallel)$ for each species, together with the various fields, are stored in the one-dimensional array *fdisi*, found in the module *DIST*. These quantities are defined at discrete grid points, so can most conveniently be thought of as multidimensional arrays of dimension 6 for the distribution function (three dimensions of space, two of velocity, and one for the species) or 3 for the fields (three dimensions of space). In the code we want to address them as multidimension arrays, via the various grid indices, in order to perform initialisation and output calculations. However, these arrays are embedded into the array *fdisi* in order to facilitate the implementation of different types of numerical schemes and optimisations.

The modules *DIST* and *INDEX_FUNCTION* are responsible for the tasks of mapping the multidimensional arrays into *fdisi* and providing a means to reference them. The *DIST* module is primarily responsible for deciding where within *fdisi* those arrays should be stored (as a contiguous block), and also making sure that each array is in a distinct part of *fdisi*. The *INDEX_FUNCTION* module determines how the indices of the multidimensional array are mapped to a single unique index for the one dimensional array. This mapping can be adjusted in order to improve parallel efficiency and (possibly) improve cache efficiency. The default ordering is at present quite optimal, but manual adjustments within the *INDEX_FUNCTION* module may yield further improvement in some cases. These two modules are quite complicated and one should look into those modules in the code or at Sect. **??** for a detailed explanation.

The function which provides the array location within *fdisi* is *indx()*. This is used in many modules throughout the code, but specifically *not* the time integration modules, which are designed not to require it. As an example of how it is used, the value *phi* of the potential at a radial grid point *ix*, a binormal grid point *imod* and an s grid point *i* is obtained in the code as

```
  phi = fdisi(indx(iphi,imod,ix,i))
```

where *iphi* is an identifier for the potential.

In addition to the local solution g and the various fields, the *fdisi_tmp* array contains parts of the solution from other processors which have been communicated via MPI datatypes. In the case of parallelising over the s-grid, the fields are also decomposed, so parts of the potential from other processors must also be present in order to calculate the derivative of $\langle\phi\rangle$. The *indx()* function deals with referencing the right part of *fdisi* so that other parts of the code do not need to make special arrangements for processor boundary points.

The distribution function f only ever exists temporarily in the array *fdis_tmp*, and in diagnostics and elsewhere **a conversion from g to f must be made (usually using the routine *get_f_from_g*, except in optimised routines) when it is necessary to work with f.**

In general, each module is responsible for allocating any arrays they require. The code attempts to do as much memory allocation as necessary early on, so that if it will exceed the memory limits of a given machine, not much time will have been wasted before the code aborts. When diagnostics are performed at the end of a run, memory allocated for time integration is freed.

8.9.4 Linear terms

The linear terms are calculated in the *LINEAR_TERMS* and *COLLISIONOP* module and put into a matrix which is in *MATDAT*. The *LINEAR_TERMS* module is designed so that more new terms can be added by only adding one new routine and calling it from within that module.

The linear terms all operate on the the distribution g (often called f in the code comments!), because the linear terms matrix is applied to **the array called *fdisi* which always contains g.**

8.10 Code parallelisation

At present, the parallelisation of the code is done primarily via decomposition of the distribution function grid and the various species in a given run, using MPI. There are now shared memory OpenMP directives implemented into the code which enable more efficient parallelisation in some cases with a large number of parallel processes. This is because performing parallel reduction operations between a large number of processes becomes expensive; at some point, when doubling the number of available processor cores, work sharing the local data between 2 cores can become more efficient than doubling the number of MPI processes. The implementation and testing of the OpenMP parallelisation can be found in Sec. 9.3.10. Here we describes the various MPI parallelisation options and their potential consequences for code performance. In the next section, a scaling test is shown for a Cray XT4.

Simple parallelisation via the fields

The equation for the distribution function (without collisions) contains no derivatives in the magnetic moment μ, which means that the distribution function for any point in the μ grid can be updated independently of the other points in the μ grid. Similarly, the distribution function for each species can be updated independently. Therefore, one can run up to *number of grid points in mu* $\times$ *number of species* parallel processes (typically 32-64 processes) to update the distribution function if parallelising over these two ‘directions’. However, these points are coupled via the fields, which must be updated before the distribution function can be calculated in an explicit time step. The field equations involve a summation over the whole of the velocity space and the species, for which the code requires an inter-process reduction sum. In parallelising over these ‘trivial’ directions, a process can first perform a partial sum over local species and μ-grid points, before participating in a sum involving all the parallel processes. For each field, this sum must be performed at every point in the 3-dimensional space, and the result must be then known to every process. As the number of utilised processor cores increases, the amount of data per core that is involved in the sum remains constant. How much the parallel efficiency of the code decreases as the number of utilised processors is increased will depend on the algorithm used by MPI_ALLREDUCE and underlying system. In this simple parallelisation case (and potentially in other cases) this ALLREDUCE operation is the bottleneck in parallel performance.

Parallelisation over the $v_\|$ grid

The equations have derivatives in parallel velocity, which in their numerical implementation involve points of the distribution function along lines in the parallel velocity grid. The derivative at the local point requires at most the value of the distribution function at 1 or 2 adjacent points (second or fourth-order scheme) in each direction along the parallel velocity grid (see Sect 8.1). When decomposing the $v_\|$ grid, the calculation of a $v_\|$ derivative at the end of a local grid will require 1 or 2 more points managed by another process. These points need to be communicated from the adjacent processor before the derivatives are performed. Typically, the 2 points at either end of the local parallel velocity grid must be communicated to the corresponding adjacent process, and 2 points must be received from it. Therefore, if we parallelise over the $v_\|$ grid we have this communication to perform in addition to that required to calculate the fields as discussed in the previous subsection. On systems which support performing computation at the same time as communication (such as a Cray XT4), this communication can be overlapped with the relatively expensive nonlinear terms computation. However, in such a case we would still be left with the same field calculation bottleneck as the number of processors responsible for the parallel velocity grid is increased.

Parallelisation over the s grid

In terms of the communication required for the derivatives along the s direction, parallelisation over the s grid is much the same as for the $v_\|$ grid. However, splitting up the s direction decomposes the space over which the fields are calculated. This means that the ALLREDUCE operation for a given processor only needs to be performed between processors responsible for the same part of the s grid. If the number of processors over which the s grid is decomposed is doubled, then the data involved in the reduction sum is halved, and the number of processors that partake in the sum remains the same. A consequence of this is that when parallelising only in the s direction, no communication is needed to calculate the fields. This is typically very favourable for parallel performance; one potential bottleneck would be effectively removed (and the other can usually be overlapped with computation).

Parallelisation over the μ (and $v_\|$) grid with collisions

If the collision operator is used, the parallelisation over the μ grid is not simply via the fields as derivatives are present. The second-order local derivative for the collision operator uses all the adjacent points in the velocity space (see Sect. 8.4). Although only 1 point must be exchanged between adjacent processors, any 1 processor may need to perform this exchange between 8 processors. If parallelising over the $v_\|$ and μ grids at the same time, this would mean communication between 6 additional processors (when the grid is decomposed over more than 2 processors in each direction) when compared to a run without collisions. Ideally we would like to avoid parallelising over both the $v_\|$ and μ grids till other options have been exhausted. Parallelising over the $v_\|$ grid, but not the μ grid should be equally expensive with or without collisions, since there is no difference in the grid points communicated. Parallelising instead over just the μ grid, but not the $v_\|$ grid would be more expensive with collisions than without, although typically cheaper (when using the default fourth-order finite-differences) than the previous case using only $v_\|$.

8.11 Parallel performance (pure MPI), Cray XT4

Here we make a simple assessment of the parallel performance of the code on a Cray XT4 (HECTOR). We define the performance as the amount of main time loop computation done by the code per unit time. In the ideal case, a doubling in the number of processor cores used will double the amount of computation done per unit time (or halve the time required to do a fixed number of time steps). We consider a non

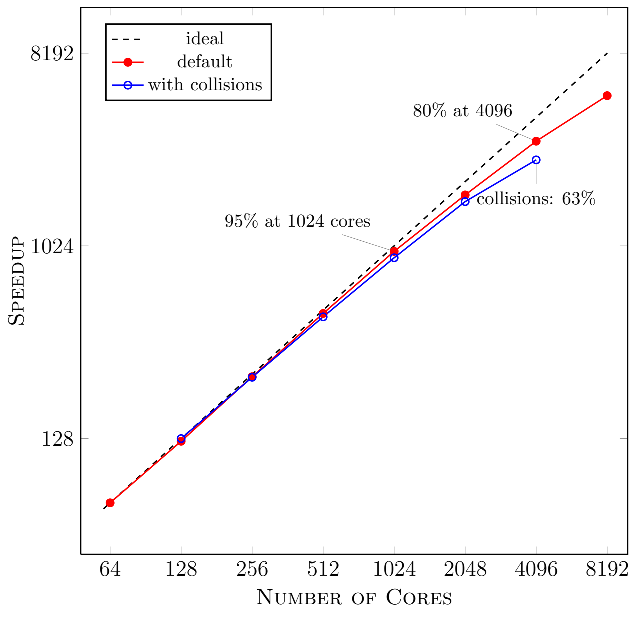


Figure 8.4: Parallel performance for a typical sized problem on HECToR (Cray XT4), both without and with the collision operator turned on. Due to machine memory constraints, this problem can run with a minimum of 64 cores without collisions and 128 cores with. The performance/speedup is based on the time taken to perform the main time integration loop of the code and is scaled such that it is equal to the number of cores when the smallest possible number of cores is used. The parallel efficiency, based on 100% for the smallest number of cores, is noted at several points on the plot.

linear problem of fairly typical size that we believe to be reasonably well converged with respect to grid resolution. As mentioned in Sect. 8.9.3, we can change the data memory layout to potentially tweak the parallelisation and cache efficiency of the code. We have experienced some variation in performance due to these adjustments. However, we do not yet have enough experience of how to optimally prescribe the layout depending on grid size and parallelisation options, so for the purpose of this test, we keep the layout fixed to the default. We perform a test with a resolution of 83 radial wave vectors, 21 bi-normal modes, 4 species, 16 s grid points, 16 μ grid points and 64 $v_\|$ grid points, using fourth-order finite-differences in double precision. Two cases are considered, one using the collision operator and one without. Due to the memory available per processor core, a minimum of 64 processors are required for the former case and 128 for the latter. Using a second-order finite-difference scheme and/or a single precision calculation can reduce this minimum. From the parallel algorithm perspective, the upper limit on the number of parallel processes used for this case is 16384. Using second-order finite differences would increase this maximum to 65536 cores. However, only up to 8192 cores were available on the hardware at the time of testing.

Before deciding how to producing a scaling plot for this testcase (such as in Fig. 8.4), some preliminary investigations on the effects of parallelising in each direction were made. Using the ‘trivial’ directions alone, the test case parallelises over the minimum 64 processors for the default case. The difference in performance between this parallelisation and parallelising primarily over the parallel velocity grid is fairly small, the ‘trivial’ directions giving less than 10% better performance than with the $v_\|$ direction. Parallelising primarily over the s grid at 64 cores (using the maximum number of cores in the s direction before using the simple direction) yields a performance which is almost identical to that using only the ‘trivial’ directions. It seems that for 64 processors, the communication needed for the derivatives has little (if any) effect on the parallel performance. To get an idea of the relative expense of parallelising over s and $v_\|$, two tests were made by starting with 64 cores (using only the trivial parallelisations), then varying either the number of processors in the $v_\|$ direction or the s direction until a total of 512 cores were used. The reduction in efficiency with increasing processors was greater with $v_\|$, with the s parallelisation performing around 6% better at 512 cores (at $>95\%$ of the efficiency at 64 processors).

A scaling test was performed for the collisionless case up to 8192 cores. Fig. 8.4 shows how the performance increases with the number of cores. For the first point at 64 cores, only the 'trivial' directions are used for the parallelisation, since that combination provides the best performance (but only by around 1%). Since the s direction is the preferred parallelisation direction (least expensive in terms of performance), the number of processors in s doubled between each point on the graph up to 512 cores, where the maximum of 8 processors in s is used. Beyond that point, the more expensive $v_\parallel$ grid is decomposed until 8192 cores are used. At 8192 cores, the code runs at 63% of the efficiency at 64 cores.

Following the discussion in the previous subsections, we should expect that with collisions, parallelisation over the μ direction is more expensive than over the s direction. Therefore, the scaling test for collisions (as shown in Fig. 8.4) starts at 128 processors fully parallelised over the s direction and over the species, using only 4 processors in μ. Then, the number of processors in μ is doubled, until 512 cores are used. At 1024 cores, it is cheaper to use 32 processors in $v_\parallel$ and only 1 in μ, since this avoids using parallelisation in both those directions together. For the final 2 points on the graph, the number of processors in the $v_\parallel$ and μ are either identical, or there are twice as many in $v_\parallel$. Since these parallelisation combinations give the best performance (considerably better than using the same ones as without collisions), careful consideration should be given to the number of processors used in each direction when using collisions. At 4096 cores the efficiency is 63% of that at 64 cores.

In both cases considered we see decreasing performance gains when we begin to parallelise in less favourable directions. Running the tests in single precision would mean less data communicated, so may improve scaling (but at the same time would likely reduce the time between communication, or the time available for overlapping communication with computation). Running with the second order finite difference scheme would allow double the number of cores in the s direction and may improve the efficiency at a large number of cores. If different test cases are considered, it may well be that more resolution in a particular direction is needed, which would then lend itself to further scaling.

For hardware in which not all processor interconnects are equal in speed, the ordering of the MPI communicators within the cartesian topology can have a significant effect on parallel performance. Current high performance machines appear with multiple shared memory cores on interconnected nodes are one example. Given the above discussion of the bottleneck in global reduction operations, it is preferable that the s direction decomposition should be over the slowest communication direction. The GKW default setup assumes that MPI rank ordering is done with adjacent physical processors adjacent in the rank ordering. Testing of a collisionless case on a Cray XT6 (HECTOR phase-2b with 24 cores per node) with 1344 cores indicated that the default dimension ordering (Species, μ, $v_\parallel$, s) was optimal. The configuration ($v_\parallel$, Species, μ, s), was 30% slower, even with exact multiples of the $v_\parallel$ grid (presumably) on a single node. This is counter-intuitive, since one would expect putting the $v_\parallel$ direction first to be optimal. Configurations with s first were up to 80% slower. The ordering can be simply reconfigured near the top of `grid.F90`. The optimal configuration will depend on hardware and problem size, and conducting tests prior to expensive runs is recommended.

8.12 Parallel performance (hybrid, MPI+OpenMP), BlueGene/P

The performance of pure MPI parallelisation has also been compared to hybrid parallelisation MPI+OpenMP (see Sec. 9.3.10 for details on the hybrid scheme) for a typical non-linear electromagnetic case with kinetic electrons. The problem size is almost the same as in the previous section: 83 radial wave vectors, 20 binormal modes, 4 species, 32 s grid points, 8 μ grid points and 64 $v_\parallel$ grid. A case without collisions and a case with collisions ($R_{\rm ref} = 3$, T_2 and $n_{\rm ref} = 6$, was investigated. In contrast to the previous section, the full collision operator *with* mass and momentum conservation was used[3].

[3]How much the scaling is affected by the communication required for the momentum conservation has not yet been fully investigated by comparisons on the same machine. Limited tests on BABEL indicated that momentum conservation caused a consistent 4% slowdown up to 4096 cores, but it could become more (or less) important at higher number of cores (note that it is overlapped with derivatives communications when the MPI implementation supports this). The version of GKW tested on BABEL did not yet have OpenMP implemented in the calculation of the momentum conserving field. Comparison of timings for the regular fields indicate that implementing this would give a 4% (2%) speedup at 4 (2) threads without MPI, but with MPI it will likely make no difference if the scaling bottleneck is in the overlapped derivatives communication.

cores	512	1024	2048	4096	8192	16384
1 thread - no OMP (time, efficiency)	203.5h 100%	212.8h 95.6%	227.9h 89.3%	255.3h 79.7%	304.3h 66.9%	408.4h 49.8%
1 thread (time, efficiency)	214.4h 100%	222.1h 96.5%	237.0h 90.5%	263.6h 81.3%	316.6h 67.7%	551.0h 38.9%
2 threads (time, efficiency)	221.1h 100%	225.5h 98.0%	233.3h 94.8%	255.1h 86.7%	313.2h 70.6%	493.1h 44.8%

Table 8.1: Case without collisions. Total main loop time (in hours) for 600 time steps and strong scaling efficiency based on the case with 512 cores.

cores	512	1024	2048	4096	8192	16384
1 thread (time, efficiency)	- (100%)	290.0h (92.6%)	317.9h (84.4%)	373.1h (71.9%)	519.2h (51.7%)	- -
2 threads (time, efficiency)	268.4h 100%	283.0h 94.8%	300.4h 89.4%	323.4h 83.0%	480.6h 55.9%	713.6h 37.6%
4 threads (time, efficiency)	281.7h 100%	283.5h 99.4%	294.9h 95.5%	314.6h 89.5%	380.5h 74.0%	506.1h 55.7%

Table 8.2: Case with collisions (including momentum conservation). Total main loop time (in hours) for 600 time steps and strong scaling efficiency based on the case with 512 cores. On 1 thread, the run at 512 cores was out of memory and the strong scaling efficiency was calculated using the results on 2 threads.

The tests were conducted on the BlueGene/P BABEL (4 cores at 850MHz and 2GB of memory per node `http://www.idris.fr/eng/User_Support/bg_support.html`) for which the number of OpenMP threads per node can be 1, 2 or 4. The executable was compiled from revision 2076 of GKW source with O4 optimisation level (typically 10% faster that O2/O3 optimisation for a compilation without OpenMP and 5% faster for a compilation with OpenMP) and double precision. Due to the small amount of memory per node available on BABEL at least 512 cores were required to run the problem under consideration (to be accurate, the run fits on 256 cores but the execution is then very slow). For all the runs in this section, the number of MPI processes in the species and s direction was *n_procs_sp*=4 and *n_procs_s*=16, respectively. The number of processes in the μ direction was *n_procs_mu*=8 for the collisionless case and *n_procs_mu*=4 for the case with collisions (except when 4 threads were used at 512 cores where *n_procs_mu* was reduced to 4 and 2, respectively). This means that the MPI parallelisation over the three most favorable directions was maximised to start with and the scaling of the least favorable direction, $v_{\parallel}$, was then compared to the OpenMP scaling by increasing either *n_procs_vpar* or the number of threads per node. For instance, a collisonless run on 2048 cores was performed on 1 thread with *n_procs_vpar*=4 or on 2 threads with *n_procs_vpar*=2. The tests were conducted with a typical set of diagnostics with NAVERAGE=100, and with the nonlinear timestep estimator on, both were found to have little effect on the scaling performance (for the single threaded case at least).

The results of the tests are summarised in Table 8.1 (no collisions) and Table 8.2 (with collisions). The corresponding speed-up and total main loop time is displayed in Fig. 8.5 and 8.6, respectively. Hybrid parallelism with 4 threads is especially useful with collisions and allows to recover the scaling obtained without collisions near the MPI limits. Compilation without OpenMP is slightly faster (due to different cache efficiencies in the matrix ordering). This was also observed on HECTOR. (This has since been changed, both now use the same sort ordering so there is no longer any cache penalty for openmp)

A test has also been performed with the number of modes increased to NX=183 and NMOD=32 with the results summarised in Table 8.3.

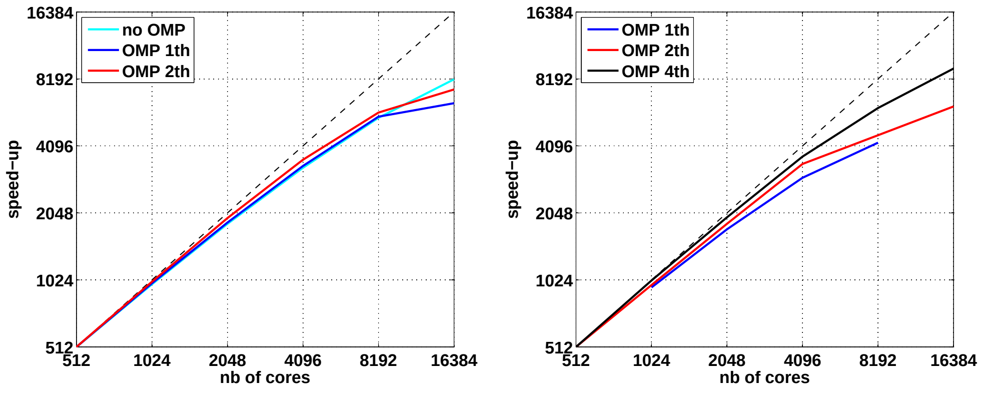


Figure 8.5: Parallel performance for a typical sized problem on BABEL (BlueGene/P), both without (left plot) and with (right plot) the collision operator (with momentum conservation) turned on. The performance/speedup is based on the time taken to perform the main time integration loop of the code and is scaled such that it is equal to the number of cores when the smallest possible number of cores is used.

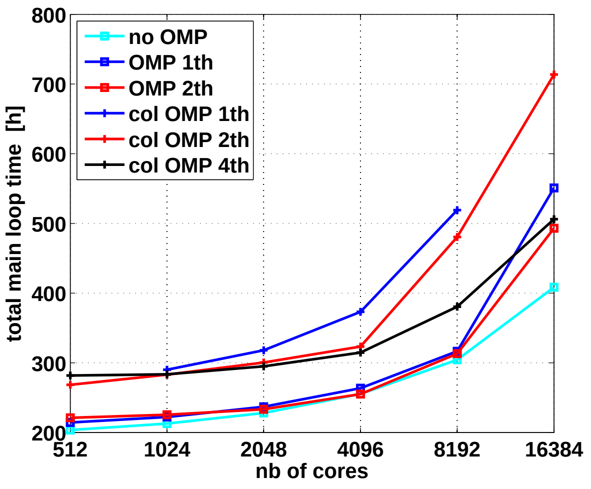


Figure 8.6: Total main loop time in hours for the cases studied in this section.

cores	2048	4096	8192	16384
1 thread (time, efficiency)	1166.4h 100%	1369.1h 85.2%	1731.7 h 67.4%	- -
2 threads (time, efficiency)	1094.9h 100%	1164.4h 94.0%	1412.5h 77.5%	2317.3h 47.3%
4 threads (time, efficiency)	1081.7h 100%	1136.5h 95.2%	1305.4h 82.9%	1692.5h 63.9%

Table 8.3: Case with collisions (with momentum conservation), NX=167 and NMOD=32. Total main loop time (in hours) for 600 time steps and strong scaling efficiency based on the case with 2048 cores.

8.13 Parallel nonspectral performance (pure MPI), Helios

The nonspectral version of the code additionally parallelizes over the radial (x) direction (with the exception an implicit solve for the polarization term). The nonspectral version of the code spends proportionally more time in the fields calculations than the spectral version, since the multi-point averages use for the gyroaverages are much slower than the Bessel functions of the spectral version. (The nonlinear terms, by contrast, are much faster since they only require 1D FFTs.)

Since the "diagonal part" of the fields solve is only 3 dimensional, it cannot parallelise over the velocity space or species. This is true in both the spectral and nonspectral versions, but in the spectral version, the diagonal part of the fields solve takes a neglible of CPU time. To allow efficient scaling for the nonspectral version, the fields solve work is additionally parallelised over the toroidal modes using the same communicators that are elsewhere used to distribute work over the velocity space and species. This feature also reduces the fraction of time spent in the implicit polarization solve, which is radially global. Two allgather communications are therefore required in the fields solve: The first in x before the polarization solve, and the second in nmod after the polarization solve so that every processor obtains the result for every toroidal mode. Since these are both gather operations of 3d quantities, they are relatively fast compared to the reductions and communications of ghost cells.

Parallelization over the x direction brings the same benefits for reducing the reduction operations in the field solve as the parallelism over the s direction. The additional penalties of extra ghost cell and allgather communications in the fields solve are outweighed by the benefit that this brings. The result is that parallelism over x is always more efficient than parallelism over parallel velocity (which in the nonspectral case performs much worse than in the spectral case due to the larger fraction of time in the field solve). For more details see doc/GKW_MPI_nonspectral.pdf.

Scaling tests of the radially nonspectral version of the code (Fig. 8.7) were performed on the Bull IFERC Helios machine (Intel Sandy Bridge with fat tree topology Infiniband communication network). As with the scaling test described in Sec. 8.11, the 'trivial' parallel directions were used first, followed by s, then x (although not to maximum), then finally vpar. Two cases were chosen to represent a medium and a large physics case simulation, with gridsizes

```
NX = 256, N_s_grid = 32,  N_mu_grid = 16,  N_vpar_grid = 32,  NMOD = 16, number_of_species = 2
```

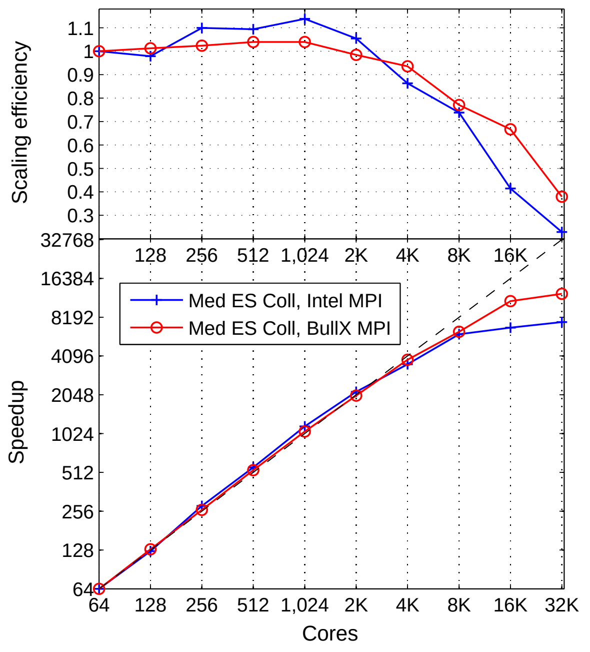
for the medium case, and

```
NX = 512, N_s_grid = 64,  N_mu_grid = 16,  N_vpar_grid = 64,  NMOD = 43, number_of_species = 2
```

for the large case.

The radial box size was `lx = 128` or `lx = 256` respectively, which resulted in 6 x ghost points required for the gyroaverages (only two are communicated for the derivatives). Since the number of ghost points must be less than the number of local points, this sets the maximum limit of `n_procs_x = 32` or 64 repsectively.

For the medium problem size, the maximum possible `n_procs_x=32, n_procs_vpar=16` would allow a theoretical maximum of 131,072 cores (65,536 with collisions). For the large problem size, the maximum possible `n_procs_x=64, n_procs_vpar=32` allows a theortical maximum of 1,048,576 cores (524,288 with collisions). These results compare favorably with the spectral scaling tests: The parallelism over x removes any scaling performance penalty associated with the use of the collision operator (which adds 20% to all runtimes).



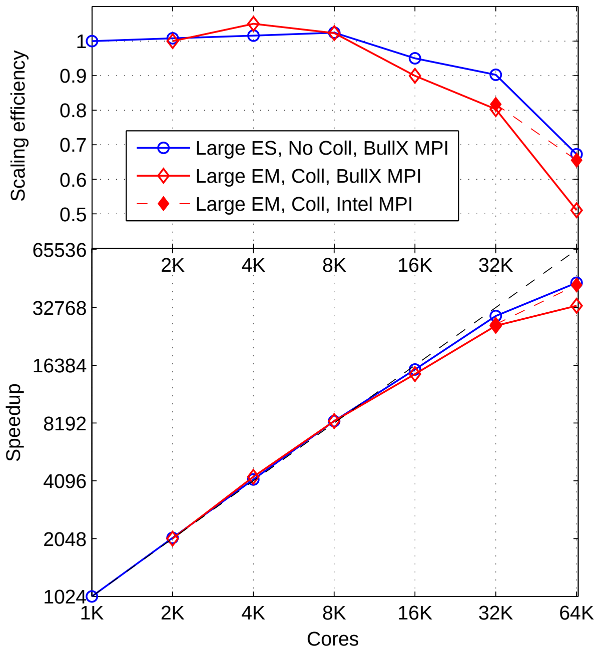


Figure 8.7: Strong scaling performance for the medium and large sized nonspectral problems on Helios. The speedup is based on the time taken to perform the main time integration loop of the code and is scaled such that it is equal to the number of cores when the smallest possible number of cores is used. BullX MPI seems to perform better at ghost cell communications of derived datatypes while Intel MPI seems to perform better at the large collective operations. Which is better therefore depends on which MPI communication is dominant for a given problem.

Chapter 9

Obtaining, building and running the code

9.1 Getting the code

To obtain the latest version of the code (which should approximately correspond to the latest version of this documentation), you should use the git revision control system (http://git-scm.com/.). The git program is installed on many systems as the *git* command. Assuming you are using *git*, you can check out the code with the command

```
git clone git@bitbucket.org:gkw/gkw.git gkw
```

To push changes, you will need a bitbucket account to be given write permissions.

9.2 Building the source code

There are two methods provided to build the code, involving various makefiles. The use of the 'simple' makefile, `src/makefile`, is intended only as a fallback if the standard makefiles, requiring *GNU make*, cannot be used. The standard GNU makefiles allow one to create build settings on a per user/machine/compiler basis, without having to make any modifications to the original makefiles[1]. Both methods are outlined below.

With the latest version of the source code, it will often be possible to build the code by doing very little other than running *make* in the top-level directory containing the `GNUmakefile` file. Although this may appear successful, one should be aware that the resulting executable may not contain all the desired functionality.

9.2.1 Prerequisites and suggestions

A Fortran 95 compiler is essential for building the code. Most Fortran compilers are able to pre-process the source files containing C pre-processor directives[2].

A minimum of a Unix-like system, together with the POSIX shell and standard *make* utility is suggested.[3]

[1] The standard makefiles also result in executables which can provide information about themselves and how they were produced. For example, the executable will be able to report its revision number in various ways, which can be quite useful to know.

[2] In the unlikely event that this is not the case, and an appropriate pre-processor is not available, some editing of the source files may be necessary before compilation.

[3] Without these basics, the various source files will have to be compiled in the correct order (or concatenated into a single file in the the correct compilation order beforehand).

For building the latest version of the code from the git repository, *GNU make* is recommended. This should be the default *make* program installed on GNU operating systems, such as GNU/Linux. On some systems, the command may be *gmake*, in order to distinguish it from the system default *make*.

To enable parallelisation, you will require an MPI implementation and/or a Fortran compiler which supports OpenMP directives. To allow nonlinear runs, the FFTW3 library is required (more details in Sect. 8.9).

A C compiler is *not* required to compile the code, but will be required to compile the optional libraries includes with GKW.

More information on linking optional libraries is given in section 9.2.6.

9.2.2 Using the GNU makefile

Instructions within the top-level makefile, `GNUmakefile`, should be the first point of reference for this method. Ultimately, you should just need to run *make* (GNU make) in this directory and you will end up with an executable in the `run/` directory. However, when building for the first time on a given machine, it is suggested to create at least one new makefile which contains settings specific to the local machine and user (and perhaps compiler if there are multiple possibilities). Whenever the code is then built again on this machine, those specific makefiles will be used automatically when running make.

All host-specific makefiles are located in the `config/` directory and subdirectories. The variables contained in `global_defaults.mk` will always be used, unless host-specific variables override them. You can use `template.mk`, which documents almost all the variables one may wish to change. If you are creating a new file for machine *hostname*, it should usually be saved to `defaults.mk` within the `hostname/` subdirectory of `config/`. You will likely want to set at least the Fortran compiler, external libraries, and the required precision. One may wish to look at existing files for examples; template files for Intel, GNU and IBM compilers are also provided in `config/templates`.

The code object files are built in different subdirectories of `/obj`, depending on hostname, compiler and precision, among other things. The resulting executable name will represent which SVN version of the code was used, which compiler was used, and if the executable is single (SP) or double precision (DP)[4]. If you run `make clean`, ALL object files will removed. `make cleanloc` will remove only object files for the compiler and precision subfolder[5]. If your make program supports it, there should be no problem in building the code in parallel (GNU make does this via the '-j' flag).

If everything completes correctly, the compiled executable will appear in the `/run` directory.

9.2.3 Using the 'simple' makefile

If using the GNU makefiles is problematic, one may wish to build the code using the simple makefile, `src/makefile`. You will need to run *make* from within the `src/` directory, but there is a good chance you will have to edit the makefile beforehand – instructions for doing this can be found in the file itself. Any files created during compilation will be in this directory, together with the source files, apart from the resulting executable, which by default, will be found in the top-level directory.

[4]Beyond these distinctions, it is possible to clobber an existing executable with one of the same name that does something different.

[5]Note that if you pass variables to the make program by specifying them after the command (rather than changing them in the appropriate file), you should be aware that existing objects might be re-used to complete the build, such that the desired properties of the program are only partly present. For example, say you compile the code without OpenMP, then edit some files, then type `make SMP=OPENMP`, then it might be the case that only some of the code has OpenMP enabled in it. To avoid such problems, use the configuration files, or type `make clean`, before (re)building to make sure you get what you expect.

9.2.4 Some important makefiles

There are two files common to both methods which are found in the `src/` directory. The `objfiles.mk` file contains the list of object files to be created before linking to produce the executable. It should only be necessary to modify this file when adding new source files to the code. The file `deps.mk` lists the dependencies for the objects found in `objfiles.mk`, so that the objects/modules may be built in the correct order. This should be updated automatically with the standard makefiles, but may need to be modified manually when using the 'simple' makefile. It *may* be possible to update it via

```
./scripts/mkdeps > deps.mk
```

from within the source directory.

The file `src/gkw.mk` is the main GNU makefile which should not need to be modified in general, unless new variables are being introduced to the code. Otherwise, all settings should be made via the files in `config/`.

9.2.5 Potential compilation issues

Most compilers tested to date have no problems compiling the code. Occasionally, some non-standard features or bugs get into the trunk version, creating problems. However, these tend to get removed rapidly after they are discovered. The Intel and GNU compilers are in most frequent use and are least likely to find unexpected new issues when compiling and running. On Hector, the PGI compiler has historically caused less problems than the Cray compiler.

Some compilers, such as those found on Fujitsu machines, may not like very large source files; if such a compiler is used, it may be necessary to edit the problematic source file and remove code which is not needed for the particular run.

Some compilers (e.g. Intel) do not by default include the present working directory in the include path. If `make` gives an error that `gkw_info.h` cannot be found, when it already exists in the `/obj` directory, you should first try adding `I./` to `FFLAGS_INC` to your `config` file.

9.2.6 Additional libraries

Here we list some third party libraries that can be used to extend the functionality of the code.

MPI-2

An MPI-2 implementation is required for any distributed memory parallel run.[6] MPI is enabled through preprocessing by setting `MPI=mpi2` in the `config` file. For most MPI implementations, the include files and linking are automatically handled by a compiler wrapper script such as `mpif90` or `mpiifort`.

FFTW3

For nonlinear runs a Fourier transform library is needed. At the moment only the FFTW3 library is supported. If this library is installed it can be linked against by setting `FFTLIB = FFT_FFTW3` in the corresponding `config` file of gkw. Depending on the system you may also need to set `FFLAGS_INC` and `LDFLAGS` to link to `-lfftw3 (-lfftw3f)` for double (single) precision.

[6]The code might compile and run (with limited MPI-I/O and nonspectral functionality) with MPI-1 but this has not been attempted for a long time, and it is likely that some MPI-2 function calls would need to be commented to make this possible.

AMD and UMFPACK

The two libraries `amd` and `umfpack` are needed for the (experimental) implicit time integration scheme and the nonspectral and global cases. They are provided with the GKW trunk and must be compiled before GKW.

At present these libraries can only be compiled in double precision. If GKW is compiled in single precision, then type conversions are made wherever these libraries are used in the code. To compile and link these libraries to GKW one must add to the GKW config file

```
IMPLICIT = umfpack      # Compile GKW with umfpack (64 bit) or umfpack32 (32 bit)
```

In most cases this should be sufficient; the libraries should then be built and linked automatically. One can also explicitly require compilation and linking of umfpack by setting

```
UMFPACK = compile       # Compile the UMFpack provided with GKW
LD_UMFPACK = -lumfpack  # link GKW with UMFpack
```

SLEPc and PETSc

For the eigenvalue solver the two libraries SLEPc and PETSc are required (where the former depends on the latter). Note that these two libraries have to be compiled for using complex numbers to work. This can be achieved with the option `--with-scalar-type=complex` for the configure script of PETSc. SLEPc inherits these settings automatically from PETSc (if configured afterwards). To link these libraries, one must add the paths for the libraries to the `LDFLAGS` together with some subset of these (-lslepc -lpetsc -llapack -lblas -lX11 -lmkl) and you must set `SLEPC = HAVE_SLEPC` in your `config` file for GKW. If the headers are not installed in a standard location for Fortran you may also have to add the library include paths to `FFLAGS_INC`. It might also be necessary to add the path to petscconf.h to `FFLAGS_INC` (probably under an `include` subdirectory for each architecture).

HDF5

The HDF5 library is needed to write output in the HDF5 data format. Note that at the moment GKW expects the HDF5 library to be compiled in its *serial* version.
To link this library, one must add the paths ot the `LDFLAGS` and `FFLAGS_INC` variables. It is a good idea to make use of the `h5fc` wrapper which is provided by the HDF5 library to make compilation easier (e.g. see the helios config). Furthermore, set `IO_LIB = HAVE_HDF5` in your `config` file for GKW, this will then enable GKW's HDF5 support.

Note that the default output format of GKW depends on the availability of HDF5. If GKW was compiled without HDF5 it then `io_format='mixed'` is the default, but if compiled with HDF5 the default becomes `io_format = 'hdf5+ascii'`, see also section 10.7.

Note also that some datasets, in particular higher-dimensional (3D+) diagnostic data (using MPI-IO) is not output in HDF5 format at the moment, because parallel HDF5 is not yet supported by GKW. If such data is requested, it will be output in fortran binary output instead of HDF5.

9.3 Running GKW

Before you do any run it is important to realize that it is not all that hard to produce nonsense. The code is not under all conditions stable, and it is not always immediately obvious if a result is wrong. It is therefore important to diagnose your runs to make sure they are O.K.. It is your responsibility as a user to produce physically meaningful results.

Here it is assumed you have the necessary MPI environment set up and running on the machine (consult your MPI library documentation) and that any run time dependencies can be satisfied.

In order to run GKW, the code requires a single input file `input.dat` to be present in the run directory. Optionally, a file containing equilibrium geometry information from the CHEASE code can be provided as input. To get started, there are a few example input files found in the input directory, together with corresponding `README` files. The file input.dat.sample lists every input switch together with a description of its purpose. input.dat.minimum is a bare minimum input file containing the essential inputs required for the code to run.

To run the code in parallel, you will probably need to use the program provided by the MPI implementation; this is usually called *mpirun* or *mpiexec*.

GKW will try to select an appropriate parallelisation given the gridsize input but this can be overridden (by setting *N_procs_[sp—mu—vpar—s]* in the gridsize namelist). Nevertheless it is sensible to have in mind a parallelisation setup when you set the gridsize, as the user controls the number of processors the code will be run on and should choose a number that fits with the gridsize. For the largest parallel runs, the user is recommended to manually choose the parallel setup.

The rest of this section describes a basic input file. Many of the input switches are optional and default values will be used if they are not provided. The input variables are provided in Fortran namelists, for example the namelist ‘control’ contains the main switches for the code. The namelist examples presented here provide you with a basic set needed to run the code. The namelists are sensitive to bad formatting, in which case the code should inform you which namelist has the problem.

GKW tries to catch as many invalid input combinations as possible. Most of the time if you input a set of incompatible switches the code will tell you and abort if necessary. For more minor errors the code will generate a warning that it is ignoring an input parameter and continue. The actual input parameters the code runs with are written to `input.out`. By renaming `input.out` to `input.dat` the code should re-run without the warnings (if the compiler correctly formats namelist outputs).

9.3.1 Modes

The spacing and location of the wavevectors is determined by the ‘mode’ namelist. For *mode_box=.false.* there is only one radial wavevector considered, but the field line can be extended to take more than one poloidal turn by setting *nperiod*. The total number of full poloidal turns is then 2 *nperiod* - 1. The bi-normal modes are determined by the input *kthrho* which specifies the value(s) of $k_\theta \rho_{\rm ref}$, see Eq. (2.256). For mode_-box=.false. the bi-normal modes are given as a list i.e. *kthrho=0.1,0.2,0.3* (the variable *krhomax* is ignored). The radial mode is determined by the input *krrho* which specifies the value of $k_r \rho_{\rm ref} = \sqrt{g^{\psi\psi}(s=0)} k_\psi$. Alternatively, a poloidal shift of the balooning angle can be specified using the input *CHIN*. A finite value of k_ψ will then be calculated to minimise $k_\perp \rho_{\rm ref}$ at the poloidal angle $\theta = CHIN$. The value of k_ψ is given by

$$k_\psi = -k_\zeta g^{\psi\zeta}/g^{\psi\psi}$$

where the metric elements are evaluated at the poloidal location specified by CHIN (poloidal angle, from the magnetic axis, positive counterclockwise, in radian and zero at the low field side midplane) and $k_\zeta = k_\theta \rho_{\rm ref}/\sqrt{g^{\zeta\zeta}(s=0)}$, as usual.

For *mode_box=.true.*, the field line must have one poloidal turn, i.e. one must set *nperiod=1*. In this case the radial modes are coupled at the end of the field line according to the ballooning shift connected with the magnetic shear. The bi-normal modes will be equally spaced from $k_\theta \rho_{\rm ref} = 0$ to *krhomax* (the *kthrho* input is ignored). The integer i_k in Eq. (2.255) is input into the code as *ikxspace* and determines the balance between radial resolution and radial box size. *ikspace* should be adjusted to give an approximately square box (the code will output LX and LY).

NS	ikxspace	NX	NPERIOD	mode_box
80	N/A	1	3	false
16	1	5	1	true
$n_s.(2n_p - 1)$	N/A	1	n_p	false
n_s	1	$2n_p - 1$	1	true

Table 9.1: Equivalent formulations example with *mode_box* true and false

9.3.2 Dissipation

The parameter *disp_vp* sets the dissipation on the trapping term. It acts a bit like a collisionality but the normalization is different: The dissipation decreases at fixed *disp_vp* when *N_vpar_grid* is increased, so that convergence should always be reached. For all collisionless problems that depend critically on the trapping (TEM modes, particle fluxes) one should not set the dissipation in velocity space too high. For these cases a lower *disp_vp* (say 0.01) will allow a lower resolution convergence of the particle fluxes. When collisions are included, *disp_vp* can and should be small (0.2).

The parameter *disp_par* sets the dissipation on the streaming term, and 0.2 is a reasonable value for most linear runs. For nonlinear runs, values in the range 0.5 - 2.0 are typically required for numerical stability, although stable results can be achieved with lower dissipation by using the Arakawa scheme. If parallel dynamics must be finely resolved, higher *N_s_grid* bay also allow lower *disp_par*. The default form of the parallel dissipation uses *idisp = 2*, which does not vary over the velocity grid and is usually more stable for electromagnetic runs . However, in some physical cases with a sharp parallel structure (usually at low k_y), excessive dissipation appears to drive a linear numerical instability; in these cases the stability may be improved by using *idisp = 1*. (See issue 201).

The circular geometry tends to be more stable than the $s - \alpha$ geometry if numerical problems arise in the zonal mode [26]. For runs with strong $E \times B$ shear, *disp_par* should be minimised, and some radial dissipation (*disp_x*=1.0) should be used (see Ref. [51] or [52]). *disp_x ¿ 0.0* is also required for the finite differences of the nonspectral version. *disp_y ¿ 0.0* can be useful in some nonlinear runs to cut out high k physical instabilities (e.g ETG modes) and supress high-k spectral pileup.

9.3.3 Gridsizes

A detailed GKW linear convergence study including many figures is available in Ref. [47], of which the numbers below are a summary. The gridsizes required for a converged result will vary for each problem, but here we give some guidance assuming *disp_vp*=0.01 and circular geometry. For adiabatic runs, $[N_s, N_\mu, N_{v_\parallel}] = [16, 8, 16]$ is often sufficient (Here, N_s is points **per poloidal turn**, but *N_s_grid* in the input file is the total number.). Collisionless ITG modes with kinetic electrons will give growth rate and frequency converged to within 5% for $[N_s, N_\mu, N_{v_\parallel}] = [16, 8, 32]$, but if the particle fluxes are to be examined, gridsizes of $[N_s, N_\mu, N_{v_\parallel}] = [40+, 8, 48]$ may be required (this is because the trapped passing boundary must be accurately resolved).

In general, trapped electron modes require more resolution to get accurate growth rate and frequencies, but once these are resolved, the particle fluxes should be also. A TEM mode is converged to with 5% for $[N_s, N_\mu, N_{v_\parallel}] = [30, 8, 64]$ Including collisions can reduce the values of $N_{v_\parallel}$ required substantially ($N_{v_\parallel} = 16 - 32$), with the lower figures for greater collisionality.

For a linear run, *Nperiod* will depend on the type of mode you are looking at. For some ITG cases with high magnetic shear, 3 would be sufficient, but for some TEM cases or cases with lower magnetic shear, as many as 10 may be required. Remember to adjust *N_s_grid* accordingly.

For a nonlinear run, *Nmod* should be at least 16. In the past a value of *nmod*=21 (or 43, 86,...) was used for particularly efficient FFT computations, but this is no longer significant. For kinetic runs *NX*=167 is a sensible size. For adiabatic runs *NX*=83 may be sufficient. For cases with very low magnetic shear or high E x B shear, *NX*=339 is sometimes used for convergence tests.

For N_s, more points are required if using complex general geometry. The boundaries of the velocity space do not usually need to be increased from the defaults.

9.3.4 Species

There is no default species data, hence the user must provide some. The species input must satisfy quasi-neutrality. This requires the user to set ratios of densities and charge to satisfy

$$\sum_s Z_s n_s = 0, \qquad \sum_s Z_s n_s \frac{1}{L_{n_s}} = 0 \tag{9.1}$$

Multiple ion species can be input but only one negatively charged species is allowed. The number of species read in is controlled by the number of species parameter in the 'gridsize' namelist and hence it is possible that later species data you specify will not be read in if this is set incorrectly. The *number_of_species* input variable does not include adiabatic electrons (the species data for the negative charge species will still be read).

Note that for a trace species, $n_s = 0$ is acceptable. The density only appears in the field equations and for scattering species in collisions. A trace particle does not contribute to the fields and does not act as a scattering species, so setting $n_s = 0$ will give the desired result.

9.3.5 Stability

The timestep required for numerical stability can vary greatly depending on input parameters, and for kinetic electrons will need to be much smaller. For a linear run, there is a time step estimate made at the start of the run and the code will post a warning if the chosen timestep is larger than this. For a non linear run, the minimum time step is estimated based on the non linear terms, then the time step is adapted as necessary. A minimum time step size can be set in the code (*dt_min*) so that the run stops when the time step becomes too small. The parallel structure of the solution is a good indicator of a the stability of a solution.

For linear runs a stable solution will converge to a constant growth rate. A tolerance for the growth rate can be specified in the code input (parameter *gamatol*), so that the run ends when the standard deviation of the growth rate computed over the last six big time steps is less than *gamatol*. When the tolerance on the growth rate is used to end the run, it is advised to keep the big time steps long enough to have a meaningful standard deviation. Typically, if the big time step length is such that *DTIM* × *NAVERAGE* > 0.2, setting *gamatol* $= 10^{-6}$ will usually ensure convergence.

GKW outputs the global growth rate in `time.dat` so a growth rate spectrum can be generated from a batch of linear runs with *NMOD=1* and *mode_box=.false.*. Typical grid sizes can be seen in the example input files.

9.3.6 β and β'

The plasma β enters the field equations (parallel and perpendicular components of the Ampere's law, Eq. 1.43 and 1.45) through the parameter β_N defined in Eq. 2.12 and recalled here:

$$\beta_N = \frac{n_{\rm ref} T_{\rm ref}}{B_{\rm ref}^2 / 2\mu_0}. \tag{9.2}$$

The parameter β_N is linked to the total plasma β at the reference magnetic field as follows:

$$\beta_0 = \sum_{\rm species} 2\mu_0 \frac{n_s T_s}{B_{\rm ref}^2} = \beta_N \sum_{\rm species} \frac{n_s}{n_{\rm ref}} \frac{T_s}{T_{\rm ref}} = \beta_N \sum_{\rm species} n_{N_s} T_{N_s} \tag{9.3}$$

The user can specify the value of β_N used by GKW within the namelist *SPCGENERAL*. Two options are implemented and controled by the switch *BETA_TYPE*:

- *BETA_TYPE='ref'*
  The value of β_N is given as an input in *BETA_REF*.
- *BETA_TYPE='eq'*.
  The total plasma β at the reference magnetic field is read from the MHD equilibrium file and β_N is calculated using Eq. 9.3 and the species parameters. The value given as an input in *BETA_REF* is not used and replaced by the calculated value. Note that this option is only available if *GEOM_TYPE='chease'* in namelist *GEOM*. If it is not the case a warning is displayed and $\beta_N = 0$ is used instead.

In addition, the radial derivative of the plasma β modifies the curvature drift, as seen in Eq. 1.21. Here again, the value at the reference magnetic field is needed (the dependence on the magnetic field is introduced directly in `linear_terms.F90`):

$$\beta_0' = \frac{2\mu_0}{B_{\rm ref}^2}\sum_{\rm species}\frac{\partial p_s}{\partial\psi} = -\beta_N \sum_{\rm species} n_{N_s}T_{N_s}[R/L_{n_s} + R/L_{T_s}] \tag{9.4}$$

Three ways of inputing β_0' are implemented:

- *BETAPRIME_TYPE='ref'*
  The value of β_0' is given as an input in *BETAPRIME_REF*.
- *BETAPRIME_TYPE='sp'*
  The value of β_0' is made consistent with the species parameters and the value of β_N defined above using Eq. 9.4. The value given as an input in *BETAPRIME_REF* is not used and replaced by the calculated value
- *BETAPRIME_TYPE='eq'*.
  The value of β_0' is read from the MHD equilibrium file. The value given as an input in *BETAPRIME_REF* is not used and replaced by the calculated value.

Note that when the species parameters are used, ALL species are considered including the adiabatic one.

Reminder: if *GEOM_TYPE='chease'*, the values of $R_{\rm ref}$ and $B_{\rm ref}$ used in GKW are the values *R0EXP* and *B0EXP* given as an input to CHEASE. It is advised that *R0EXP* and *B0EXP* are chosen to be the radius of the magnetic axis and the value of the magnetic field at this position respectively for better convergence in CHEASE.

9.3.7 Generating a linear growth rate spectrum

GKW currently outputs the global growth rate (and frequency) in `time.dat` so the easiest way to generate a growth rate spectrum is do a batch of linear runs with *NMOD=1*, *mode_box=.false.*, and *NX=1*, and scan over *kthrho* and take the growth rate from each. The matlab scripts create_gkwscan.m and growth_rate_spec.m help with creating the input and plotting the output; script gkwnlin runs the batch jobs.

Alternatively, a growth rate (frequency) spectrum can be generated by running with *mode_box=.true.* and setting *lgrowth_rates=.true.* (*lfrequencies=.true.*)in the *DIAGNOSTIC* namelist. For each toroidal mode, the growth rate and of the mode with $k_\psi = 0$ will be output. To get this spectrum to exactly match a spectrum generated by the first method at higher k_ζ will require large *NX* (and low *ikxspace*) to fully resolve the mode along the field line. The excercise of matching the spectra generated by these two methods will require the user to understand the mode coupling over the parallel boundary condition and will indicate the *nx* resolution required for *mode_box=true* and nonlinear runs.

9.3.8 Restarts and checkpoints

On a successful exit, GKW writes a binary file `FDS` which contains the distribution function needed for a restart. Along with the input file, this is the only file required to restart the code. For seamless continuation, some other quantities such as file counters and normalized time are also written to the ASCII file `FDS.dat` which contains Fortran namelist which the code reads. The binary restart file is written using MPI-IO and is independent of processor layout. This means that runs can be restarted with different parallelizations from the original run.

GKW will only attempt to restart from the above files if *read_file=T* in the *CONTROL* namelist. If it does not find any files, it will initialize a new run. Under the default behaviour, when this subsequent run completes, a restart files `FDS` and `FDS.dat` will be overwritten. This behavior can be altered by setting the integer *irun* in the *CONTROL* namelist which allows incremental numbering of restart files. If *irun=1*, `FDS` is read if present and `FD1` is written. If *irun=2*, `FD1` is read and `FD2` is written, and so on.

For systems on which MPI-IO is not available or problematic[7], GKW can also be made to write a legacy restart file format, in which each processor writes its own file `FDSXXXXXX` numbered by processor (option *restart_file_version=1* in *CONTROL*). In this case, restarts may not alter the parallelization.

Intermediate checkpoints can also be written by setting the integer *n_dump_ts=50* in the *CONTROL* namelist, to write, for instance a checkpoint every 50 large timesteps. This is strongly advised for any expensive nonlinear simulation. Two restart files `DM1/2` are alternately overwritten by a newer one each time. These files have exactly the same format as the final restart file `FDS` and can be used for a restart by simple renaming. One should note that if restarting an interrupted run from a checkpoint, some the time trace files may have duplicated lines, which can be easily verified by checking or plotting column 1 of `time.dat`.

A clean exit can be requested at anytime while the code is running by creating a file named *gkw.stop* in the run directory. An additional checkpoint file can be created by instead creating a file named *gkw.dump*, after which the code will continue to run.

Some batch systems will automatically resubmit a job which is terminated by a hardware problem; to allow GKW to automatically restart from the `DM1/2` files (and take precedence over FDS files) if present, set *auto_restart=T* in the *CONTROL* namelist. GKW could also be made to interface with batch systems which are provide accounting flexibility for codes able to halt on request, by means of the *gkw.stop* file creation mechanism.

[7] For large problems with more than about 64 processes, there can be problems/errors when reading/writing the restart files in parallel, depending the filesystem and the MPI implementation. This has been observed on some Cray XT4 machines, where in some cases it was necessary to modify environment variables to successfully write restart files. There is also a workaround in the code which must be set at compile time via adjustments to two variables in `switches.F90`: Rather than writing to file with all processes simultaneously, this allows writes with subsets of processes at a time, where the file is closed then re-opened between subsets. Increasing the number of subsets reduces overheads and can allow files to be written successfully. This might also improve the I/O performance on machines, where such operations already work. However, if the number of subsets is too large, the I/O performance will suffer. The variable *min_fdisi_io_subsets* selects the minimum number of process subsets to use. When problems are encountered, it is suggested to first try setting this value to the value of *n_procs_sp*, then to *n_procs_sp*n_procs_s* if problems persist. Using larger values will likely give very poor performance, but may be necessary. The other variable, *min_procs_io_split*, sets the minimum number of processes for which the read/write split should be used.

9.3.9 Tips and tricks

This section aims to document other useful things to know when running the code. Please add here anything else useful to the user that we have yet to document properly. Received wisdom, etc.

- The linear timestep estimator is based on the Courant condition for the Vlasov equation, and the frequency of shear-Alfvèn wave [70], so it is not foolproof for some high frequency waves (e.g. ETG). If GKW gives unphysicaly large growth rates, the first thing one should try is to reduce the timestep (no more than a factor of two at a time). If your problem requires a very small timestep (less than 1e-6), something else is likely to be wrong, but if a very small timestep is really required it may be best to run with *normalized=.false*. Equally, for most efficient resource usage, one should try to find the upper limit of the stable timestep.
- For nonlinear kinetic runs, adding a small electromagnetic component e.g. $\beta = 0.03\%$ can stabilise some long wavelength instabilities (electrostatic shear-Alfvèn waves), allowing a much larger stable timestep, as suggested in [71].
- The nonlinear timestep estimator rarely manages to prevent blowup in the event of numerical instability, but can be useful for strongly nonlinear physics. For very large numbers of processors it may cause an overhead which reduces the scaling efficiency. It can therefore be disabled by setting. *nl_dtim_est=F* in the *CONTROL* namelist, which will cause the code will abort rather than continue if the timestep estimate is violated. The code could then be restarted from the previous dumpfile with the nonlinear timestep estimator enabled, or with a smaller timestep.
- The nonlinear timestep estimator is often too conservative. If your problem is limited by the nonlinear timestep, it may be possible to save CPU resources by increasing it (no more than 90%) by setting e.g. *fac_nl_est = 1.6* in *CONTROL*.
- Don't be tempted to use more cores than you need. Pushing the parallelisation to its limit (particularly less than 4 points per processor in parallel velocity) will result in inefficient scaling.
- Be aware of the effects of the boundary condition at the end of the field line. If not running a box of modes, you should check that *nperiod* is large enough so that the results are insensitive to the free boundary. If running with *mode_box*, care should be taken with the radial mode resolution so that most toroidal modes have several periods and do not feel the effect of the boundary. This is particularly important when kinetic electrons are used.

9.3.10 Hybrid scheme

Hybrid parallelism combining distributed memory MPI parallelism with shared memory OpenMP parallelism has been implemented to allow the code to scale further, and more efficiently. Many modern high performance machines now have an architecture with a number of shared memory CPU cores on each node (2-24), and a large number of these nodes connected by a fast network interconnect. The communication between nodes is much slower than between cores, and a well optimised and well utilised MPI implementation will take advantage of this by putting the most intensive communication on the local node. In cases where this is not done, or cases where a subset of the problem grid does not fit nicely on a single node, the efficiency of the inter-node communication may be increased by reducing the number of MPI processes involved in the intra-node communication. For example, on a machine with 8 cores per node (e.g. HPC-FF), 1024 processors can be utilised with the combinations (number of MPI processes, number of OpenMP threads per node) = (1024,1), (512,2), (256,4), or (128,8), where the lower number of MPI processes in the latter case may result in a more efficient reduction sum (though in theory the amount of inter-node data communicated by the ideal MPI implementation would remain the same).

In GKW, hybrid parallelism (for the spectral scheme only) is achieved by subdividing the most CPU intensive loops amongst OpenMP threads. The slowest of these is the loop in nonlinear terms, where for each local grid point, 5-7 FFTS in 2D must be performed. At the maximum limit on the number of MPI processes, there will always be at least 2 (4 with collisions) points in this loop (it contains *NMU*NSP*NS* local grid points) which can be divided between threads. (The code could be rewritten here to increase the number of threads that can be used in this section, if the hybrid scheme is demonstrated to bring a clear benefit at these numbers of threads). The second largest loop which must be parallelised is the sparse matrix-vector multiply involved in the linear terms. Due to the sparse matrix storage format, this loop cannot be vectorised, but cache efficiency can be maximised by an appropriate sorting of the matrix. After this sort, the matrix elements are subdivided between threads at carefully chosen boundaries, to ensure there is no data race in the update of the right hand side. The same technique is used for the fields calculation. All possible smaller loops (<5% runtime) are also parallelised with OpenMP, but here there are diminishing returns here due to the overhead involved in thread branching. This inherently serial element of the code will set the limit of the number of threads that can efficiently be used (likely four at present).

The hybrid scheme with OpenMP parallelism can allow GKW to scale better in certain situations for large runs. Tests on HPC-FF and Babel Bluegene machines (combined with some reasoning), suggest that the cases in which it is most likely to be useful are those in which the MPI communications are the bottleneck. When this occurs depends on the problem and the hardware. These situations are likely to include one or more of the following:

- Problems with large numbers of modes
- Problems with collisions (4 threads has been demonstrated to bring a benefit here)
- Runs on very large numbers of processsors, near the limit of GKW MPI capability
- Hardware with slow communication (HPC-FF or BlueGene as opposed to Hector)

Preliminary tests with the hybrid scheme on up to 1536 processors (on HPC-FF: IBM-Intel-Xeon-Infiniband) demonstrated up to a 5% performance increase using two OpenMP threads per node. More extensive scaling tests on Bluegene/P BABEL have demonstrated a scaling efficiency increase of up to 20% with 4 threads when running a collisional case on 8192 cores and above (for more details see Sec. 8.12).

You are advised to do some brief checks for your problem and hardware to determine if the hybrid scheme is an advantage. Best results will probably be obtained using the hybrid scheme with no more than four OpenMP threads per MPI process. For acceptable results, the number of threads must be a factor of the local processor gridsize *NMU*NSP*NS* (which is also the maximum number of threads), as a loop of this size is parallelised in the nonlinear terms.

To use the hybrid scheme, you need a compiler that supports OpenMP. You will need to compile with SMP=OPENMP preprocessor flag, openmp compiler flag, and link with the correct libraries (by using the same compiler flag). In the standard makefile build process, this is usually achived by setting

```
SMP=OPENMP
FFLAGS_OMP=-openmp (for intel, or -fopenmp for gnu)
```

Using the compile optimisations specific to the processor is also recommended.

9.4 Example GKW input file

Further example input files are provided with the code, including a comprehensive input file input.dat.sample, a minimal input input.dat.minimum, and input files for some common benchmark cases in input/.

```
&CONTROL                   !Namelist read in module CONTROL
 non_linear = .false.      !Include the non linear terms, computationally expensive fft
 zonal_adiabatic = .false.,!If zonal flows correction included for adiabatic electrons
 order_of_the_scheme = 'fourth_order'  !or 'second_order' in space
 dtim   = 0.02          !Time step size (normalised time)
 ntime  = 25,           !Number of iterations of NAVERAGE.
 naverage = 100,        !Number of timesteps between re-normalisation, data output
 nlapar = .false.,      !true = keep parallel electro-magnetic potential.(electrostatic=false)
 nlbpar = .false.,      !true = keep compressional magnetic field (electrostatic=false)
 collisions = .false.   !Turn on the collision operator
 disp_par = 1.0         !Dissipation coefficient for parallel derivatives.
 disp_vp  = 0.0         !Dissipation coefficient for parallel velocity space
 max_seconds = 1000.    !Internal program kill time.
 gamatol     = 0.       !tolerance in the linear growth rate for which the code should stop
 io_format   = 'ascii'  !chose output format to be ascii
/
&GRIDSIZE        !Namelist read in module GRID
 nx = 83,        !Number of radial wave vectors - needs to be an odd number for fft
 n_s_grid = 50,  !Number of grid points along the field line
 nperiod = 1,    !Integer - the field line makes 2*nperiod - 1 poloidal turns.
 n_mu_grid   = 8,        !Number of magnetic moment grid points
 n_vpar_grid = 32,       !Number of grid points for parallel velocity
 nmod = 21,              !Number of bi-normal modes (k_zeta)
 number_of_species = 1   !Number of kinetic species.
/
&MODE                !Namelist read in module MODE
 kthrho  = 0.5,1.0,  !'Poloidal' wavevector(s) if mode_box=.false.
 mode_box = .true.,  !Determines if there is a 2D grid of modes. If true use nperiod = 1
                     !Also determines if radial modes are coupled at end of field line.
 krhomax = 1.0,      !For mode_box, the maximum 'poloidal' wavevector (k_perp rho_i) used
                     !k_perp evaluated on the low field side of the outboard midplane
                     !rho_i evaluated on the flux surface at the major radius of magnetic axis
 ikxspace = 7,       !Spacing between radial modes (resolution against box size)
/
&SPCGENERAL                     !Namelist read in module COMPONENTS
 beta_ref = 0.000,              !Plasma beta (not used if both nlapar, nlbpar are .false.)
 adiabatic_electrons = .true. !Kinetic electrons require a smaller timestep
/
&SPECIES       !Namelist read in module COMPONENTS
 mass  = 1.0,  !Species mass in terms of reference mass
 z     = 1.0,  !Species charge. If negative, assumed to be the electrons.
 temp  = 1.,   !Temperature of species scaled by reference temperature
 dens  = 1.,   !Density of species scaled by reference density
 rlt   = 8,    !Temperature gradient R/LT
 rln   = 3.5,  !Density gradient R/Ln
 uprim = 0.0,  !Gradient of the toroidal rotation Rref^2 grad Omega / v_thref
/
&SPECIES
 mass  = 2.72e-4, !Electrons
 z     = -1.0,    !Only one negatively charged species is allowed.
 temp  = 1.0,
 dens  = 1.0,
 rlt   = 7.9,
 rln   = 3.5,
 uprim = 0.0,
/
&GEOM          !Namelist read in module GEOM, s-alpha is default geometry
 shat = 1.0,   !Magnetic shear
 q    = 2.0,   !Safety factor
 eps  = 0.1    !Radial Coordinate
/
&ROTATION                    !Namelist read in module ROTATION
 vcor = 0.0                  !Rotation of the plasma vcor = Rref Omega / vthref = \Omega_N
 shear_rate = 0.0            !Normalised shearing rate for the E x B perpendicular shear
 shear_profile  = 'none'     !To include a shear flow, use `wavevector_remap'
 cf_trap = .true.            !Include the centrifugal trapping effects
 cf_drift = .true.           !Include the centrifugal drift
/
&COLLISIONS    !Namelist read in module COLLISONOP if collisions = .true.
 rref = 1  !reference major radius used in collision operator
 tref = 1  !reference temperature in units of kev used for the collision operator
 nref = 1  !reference density in units 10^19 m^-3 used for the collision operator
 mom_conservation = .false.     !Use the correction to conserve momentum
 mass_conserve = .true.,        !Forces zero flux boundary on velocity space so no out flow of mass
 freq_override=.false.          !Manually specify ii collision frequency to override ref values
 coll_freq=6.515E-5             !Ion-ion collision frequency used if freq_override=.true.
/
```

Chapter 10

Diagnostics

In GKW, each diagnostic constitutes a separate module `diagnos_foobar`. The distinction between modules is mainly made by the physical quantities they compute - and usually not their dimensionality.

A template with some typical code snippets and comments on writing new diagnostics can be found in `diagnos_template.F90`.

10.1 Fluxes (module `diagnos_fluxes`)

The fluxes are calculated as guiding centre fluxes (flux surface averaged) and are given by the equations

$$\{\mathbf{\Gamma}_i \cdot \nabla\psi\} = \Gamma_i^{\psi} = \left\{\int \mathrm{d}^3\mathbf{v}\tilde{\mathbf{v}}_E \cdot \nabla\psi\alpha_i f\right\} + \left\{\int \mathrm{d}^3\mathbf{v}\tilde{\mathbf{v}}_{\delta B} \cdot \nabla\psi\alpha_i f\right\} + \left\{\int \mathrm{d}^3\mathbf{v}\tilde{\mathbf{v}}_{\nabla B_{1\parallel}} \cdot \nabla\psi\alpha_i f\right\} \tag{10.1}$$

for the electro-static and electro-magnetic (due to magnetic flutter and compression) fluxes, respectively. In the above equation all quantities are dimensional and unnormalised (with the exception of the dimensionless coordinate $\psi = \epsilon$), and $i = (1, 2, 3)$ with $\alpha_i = (1, v^2/2, \frac{s_{\mathrm{B}} R B_t}{B} m v_\parallel)$, i.e. $i = 1$ is the normalised particle flux, $i = 2$ is the heat flux, $i = 3$ is the toroidal angular momentum flux (with the toroidal flow counted positive in the toroidal direction, clockwise from above). The toroidal angular momentum is the quantity that is conserved in tokamaks, and toroidal angular momentum flux is what enters the 1D transport equation. This flux is positive for outward flux of plasma rotating in the toroidal direction (clockwise from above), which differs from the parallel direction when $s_B = -1$.

Both the potential and the distribution function are represented as a sum over Fourier modes. Since the flux contains an average over space it will be zero unless the wave vectors of the potential and distribution function add up to be zero. In the representation of modes used in this manuscript this means that a particular wave vector of the representation of f must be combined with the complex conjugate of the same wave vector in the representation of ϕ or $A_\parallel$ (sec. 10.4), i.e.

$$\langle \hat{v}_E f\rangle = 2\sum_m \mathrm{Re}[\langle \hat{\mathbf{v}}_{Em}^{\dagger} \cdot \nabla\psi \hat{f}_m\rangle], \tag{10.2}$$

where $\sum_m = \sum_{k_{\zeta,\min}}^{k_{\zeta,\max}} \sum_{-k_{\psi,\max}}^{k_{\psi,\max}}$ is the sum over the modes as kept in the code (noting that zero-mode does not contribute to the fluxes), $\hat{v}_{Em}$ and $\hat{f}_m$ are the Fourier amplitudes of the $E \times B$ velocity and distribution function of the mode m respectively, and the dagger denotes the complex conjugate. Substituting the various definitions one can derive the expression for the normalised fluxes (electro-static and electro-magnetic)

$$\mathcal{I}_i = \sum_m \left\{ 2\mathcal{E}^{\psi\beta} k_{\beta m} \int 2\pi B \, \mathrm{d}\mu \mathrm{d}v_\parallel \, \hat{\alpha}_i \mathrm{Im}[\langle \hat{\phi}_m \rangle^\dagger \hat{f}_m] \right\} = \sum_m \left\{ \bar{\mathcal{I}}_i(k_\psi, k_\zeta, s) \right\}, \tag{10.3}$$

$$\mathcal{J}_i = -2 \sum_m \left\{ 2\mathcal{E}^{\psi\beta} k_{\beta m} \int 2\pi B \, \mathrm{d}\mu \mathrm{d}v_\parallel \, \hat{\alpha}_i v_R v_\parallel \mathrm{Im}[\langle \hat{A}_{\parallel m} \rangle^\dagger \hat{f}_m] \right\} = \sum_m \left\{ \bar{\mathcal{J}}_i(k_\psi, k_\zeta, s) \right\}, \tag{10.4}$$

$$\mathcal{K}_i = 2 \sum_m \left\{ 2\mathcal{E}^{\psi\beta} k_{\beta m} \int 2\pi B \, \mathrm{d}\mu \mathrm{d}v_\parallel \, \hat{\alpha}_i \frac{\mu T_R}{Z_{sp}} \mathrm{Im}[\langle \hat{B}_{\parallel m} \rangle^\dagger \hat{f}_m] \right\} = \sum_m \left\{ \bar{\mathcal{K}}_i(k_\psi, k_\zeta, s) \right\} \tag{10.5}$$

where here all quantities are normalised and $\sum_m \{\} = \sum_{k_\zeta} \sum_{k_\psi} \int_s \mathrm{d}s$ is a sum over all modes and a flux surface average over s, and where $\hat{\alpha}_i = (1, v^2, \frac{s_\mathrm{B} R B_t}{B} v_\parallel)$. (In the code, the factor 2π from the velocity space Jacobian is included in the array *intmu*=$2\pi\Delta\mu$).

The fluxes in physical units can be obtained from

$$R_\mathrm{ref}\Gamma_s^\psi = n_{R_0,s}\rho_*^2 v_\mathrm{thref}(\mathcal{I}_1 + \mathcal{J}_1 + \mathcal{K}_1), \tag{10.6}$$

$$R_\mathrm{ref}Q_s^\psi = n_{R_0,s}T_s\rho_*^2 v_\mathrm{thref}(\mathcal{I}_2 + \mathcal{J}_2 + \mathcal{K}_2), \tag{10.7}$$

$$R_\mathrm{ref}\Pi_{\phi s}^\psi = m_s n_{R_0,s} v_{ths} R_\mathrm{ref}\rho_*^2 v_\mathrm{thref}(\mathcal{I}_3 + \mathcal{J}_3 + \mathcal{K}_3), \tag{10.8}$$

for particle, heat, and angular momentum fluxes respectively ($\Gamma_s^\psi, Q_s^\psi, \Pi_{\phi s}^\psi$ are the contravariant components of the flux and therefore have units of particle, heat, angular momentum flux per unit length respectively). Note: In some GKW publications the flux of parallel momentum $\Gamma_\parallel$ has been used. This is meaningful only for the $s-\alpha$ geometry in which the finite ϵ terms are neglected, in which case $\Pi = R_\mathrm{ref}\Gamma_\parallel$.

Note that when calculating momentum diffusivity, the definition of u' relative to v_thref means that the factor $v_R = v_{thi}/v_\mathrm{thref}$ must be inserted if $T_\mathrm{ref} \neq T_i$. The Prandtl number (for example) is therefore calculated as

$$Pr = \frac{\mathcal{I}_3}{\mathcal{I}_2}\frac{R/L_T}{u'} v_R. \tag{10.9}$$

The suprema of the fluxes (for which we use the notation $\hat{\mathcal{I}}_i, \hat{\mathcal{J}}_i, \hat{\mathcal{K}}_i$), are defined the same way as the fluxes in equation 10.3 etc, but replacing $\hat{f}_m \rightarrow -i|\hat{f}_m|$ and $\hat{\phi} \rightarrow |\hat{\phi}|$. The fluxes suprema represent the largest possible fluxes for the fluctuation amplitudes, which would occur if the fluctuations were out of phase by $\pi/2$. The ratio between the fluxes and the flux suprema

$$\frac{\mathcal{I}_i}{\hat{\mathcal{I}}_i} = \sin\{\hat{\theta}\} \tag{10.10}$$

then contains the cross phase information, where $\{\hat{\theta}\}$ is the average phase angle between the fluctuating quantites in the flux (note, that from this ratio one cannot distiguish phase angles outside the principal value).

10.2 Fluxes in more detail (module `diagnos_fluxes_vspace`)

There is the possibilibty to output the unintegrated fluxes by setting the switch *lfluxes_detail*, with contributions of all field fluctuations which are considered by the run.

$$\bar{\mathcal{I}}_i(k_\psi, k_\zeta, s, \mu, v_\parallel) = 2\mathcal{E}^{\psi\beta} k_{\beta m} 2\pi B \, \mathrm{d}\mu \mathrm{d}v_\parallel \, \hat{\alpha}_i \mathrm{Im}[\langle \hat{\phi}_m \rangle^\dagger \hat{f}_m] \mathrm{d}^3 X \tag{10.11}$$

$$\bar{\mathcal{J}}_i(k_\psi, k_\zeta, s, \mu, v_\parallel) = -2 \cdot 2\mathcal{E}^{\psi\beta} k_{\beta m} 2\pi B \, \mathrm{d}\mu \mathrm{d}v_\parallel \, \hat{\alpha}_i v_R v_\parallel \mathrm{Im}[\langle \hat{A}_{\parallel m} \rangle^\dagger \hat{f}_m] \mathrm{d}^3 X \tag{10.12}$$

$$\bar{\mathcal{K}}_i(k_\psi, k_\zeta, s, \mu, v_\parallel) = 2 \cdot 2\mathcal{E}^{\psi\beta} k_{\beta m} 2\pi B \, \mathrm{d}\mu \mathrm{d}v_\parallel \, \hat{\alpha}_i \frac{\mu T_R}{Z_{sp}} \mathrm{Im}[\langle \hat{B}_{\parallel m} \rangle^\dagger \hat{f}_m] \mathrm{d}^3 X \tag{10.13}$$

10.3 Growth rates, Frequencies (module `diagnos_growth_freq`) and Quasi-linear fluxes

For a linear run, the dominant (overall) mode growth rate is calculated as

$$\gamma^{\max}(t) = \ln\left[\frac{\sqrt{\sum_m \int |\hat{\phi}(t)|^2 + |\hat{A}_{\|}(t)|^2 + |\hat{B}_{\|}(t)|^2 \,\mathrm{d}s}}{\sqrt{\sum_m \int |\hat{\phi}(t-\Delta t)|^2 + |\hat{A}_{\|}(t-\Delta t)|^2 + |\hat{B}_{\|}(t-\Delta t)|^2 \,\mathrm{d}s}}\right] / \Delta t. \tag{10.14}$$

The corresponding total mode frequency is calculated as

$$\omega(t) = \left[\mathrm{Arg}\left(\sum_m \int \hat{\phi}(t) + \hat{A}_{\|}(t) + \hat{B}_{\|}(t)\mathrm{d}s\right) - \mathrm{Arg}\left(\sum_m \int \hat{\phi}(t-\Delta t) + \hat{A}_{\|}(t-\Delta t) + \hat{B}_{\|}(t-\Delta t)\mathrm{d}s\right)\right] / \Delta t. \tag{10.15}$$

These values are output in columns 2 and 3 of `time.dat` respectively. Positive values of $\omega(t)$ indicate propagation in the direction of $B \times \nabla B$ (opposite to $\nabla\zeta$), which for $s_{\mathrm{j}} = 1$ and $R/L_n > 0$ is in the ion diamagnetic drift direction. In a run with *mode_box=F* (including nonlinear runs), per mode growth rates and frequencies are also produced in a similar way in the files `growth.dat` and `frequencies.dat`. For these diagnostics, however, only the k_ψ modes connected to the $k_\psi = 0$ by the condition Eq. 2.254 mode are included in the sum, which makes these mode properties directly comparable to a run with *mode_box=F* and θ_0=*CHIN=0*.

In the case of runs with *mode_box=T*, the values above (always written to `time.dat`) correspond to a global growth rate and frequency summed over all modes, which usually corresponds closely to the fastest growing mode. In addition, the growth rate and frequencies for each individual mode connected to $k_\psi = 0$ are also calculated and written to the files `growth.dat` and `frequencies.dat`.

For a normalised linear run, the solution is normalised by

$$\sqrt{\sum_m \int |\hat{\phi}(t)|^2 + |\hat{A}_{\|}(t)|^2 + |\hat{B}_{\|}(t)|^2 \mathrm{d}s} \tag{10.16}$$

which means that the outputs in `fluxes.dat` correspond to

$$(\mathcal{I}_i, \mathcal{J}_i, \mathcal{K}_i) / \left(\sum_m \int |\hat{\phi}(t)|^2 + |\hat{A}_{\|}(t)|^2 + |\hat{B}_{\|}(t)|^2 \,\mathrm{d}s\right) \tag{10.17}$$

and can therefore be used directly in Quasi-linear models. When comparing with nonlinear or unnormalised runs (at least in the linear phase), these Quasi-linear normalised values in `fluxes.dat` should be directly comparable to the values in the fluxes spectra divided by the sum of the per mode values in the files `kyspec` and `kyspec_em`.

10.4 Hermitian symmetry of binormal spectra

Many diagnostics calculate quantities which are quadratic in the perturbations and output them either as k_ζ spectra or integrated over k_ζ. According to what is often called Parseval's Theorem one can integrate the Fourier-space field.

$$\begin{aligned} \int \mathrm{d}\zeta |f(\zeta)|^2 &= \int \mathrm{d}k_\zeta |\hat{f}(k_\zeta)|^2 \\ \int \mathrm{d}\zeta f(\zeta)\phi(\zeta) &= \int \mathrm{d}k_\zeta \hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta) \end{aligned} \tag{10.18}$$

The Fourier representation of a real-valued field possesses Hermitian symmetry, as

$$\hat{f}(k_\zeta) = \frac{1}{2\pi}\int \mathrm{d}\zeta f(\zeta) e^{ik_\zeta \zeta} \tag{10.19}$$

one finds

$$\hat{f}(-k_\zeta) = \frac{1}{2\pi}\int \mathrm{d}\zeta f(\zeta) e^{i(-k_\zeta)\zeta} = \hat{f}^*(k_\zeta) \qquad \text{and also} \quad \hat{f}(-k_\zeta,-k_\psi) = \hat{f}^*(k_\zeta,k_\psi) \tag{10.20}$$

This symmetry allows the FFT libraries and GKW to save unnecessary memory for k_ζ-space quantities. For a position space grid having *n_y_grid* elements, the Fourier space representation has in principle also *n_y_grid* coefficients, but due to the symmetry, only *floor*$(\frac{\textit{n\_y\_grid}}{2}) + 1 =$ *nmod* of them are stored in memory (Note that in contrast to this, radially spectral quantities have *n_x_grid* elements). When it comes to integration, the sum $\int \mathrm{d}k_\zeta$ is to be taken over all *n_y_grid* coefficients, however.

$$\begin{aligned}
\int \mathrm{d}k_\zeta \hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta) &\hat{=} \hat{f}^*(0)\hat{\phi}(0) + \sum_{-k_{\zeta,max}}^{-\Delta k_\zeta} \hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta) + \sum_{\Delta k_\zeta}^{k_{\zeta,max}} \hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta)\\
&\hat{=} \hat{f}^*(0)\hat{\phi}(0) + \sum_{k_{\zeta,max}}^{\Delta k_\zeta} \hat{f}^*(-k_\zeta)\hat{\phi}(-k_\zeta) + \sum_{\Delta k_\zeta}^{k_{\zeta,max}} \hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta)\\
&\hat{=} \hat{f}^*(0)\hat{\phi}(0) + \sum_{k_{\zeta,max}}^{\Delta k_\zeta} \left(\hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta)\right)^* + \sum_{\Delta k_\zeta}^{k_{\zeta,max}} \hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta)\\
&\hat{=} \hat{f}^*(0)\hat{\phi}(0) + \sum_{k_{\zeta,max}}^{\Delta k_\zeta} 2\Re\left\{\hat{f}^*(k_\zeta)\hat{\phi}(k_\zeta)\right\}
\end{aligned} \tag{10.21}$$

Hence an integral over a modulus square has to be computed as

$$\int \mathrm{d}k_\zeta |f(k_\zeta)|^2 \hat{=} \hat{f}^*(0)\hat{f}(0) + \sum_{k_{\zeta,max}}^{\Delta k_\zeta} 2|\hat{f}(k_\zeta)|^2 \tag{10.22}$$

To simplify the code and to make the factor 2 explicit, the array *parseval_correction* was introduced. Here is an example of how it is used:

```
real :: integral = 0.0
integer :: ix = 1, i = 2, j = 3, k = 4, is = 1
! integrate |f|**2 over zeta, with all other coordinates fixed to some value
do imod = 1, nmod
  integral = integral + parseval_correction(imod) *
               abs(get_f_from_g(imod,ix,i,j,k,is,fdisi))**2
end do
```

Several diagnostics output binormal spectra. Note that some of these contain the factor 2 (e.g. the entropy *entr* from *diagnos_energetics*), while others do not (e.g. the intensity spectrum *kyspec* from *diagnos_fields*). While for some considerations the factor 2 in front of the non-zonal-mode contributions may not matter much for the qualitative picture, it is usually important to consider it if the quantity has to fulfill certain consistency conditions.

10.5 Tips for unit conversions

Although GKW uses the thermal velocity ($\sqrt{2T/m}$) for normalisation, it is common in the literature to use the gyro-Bohm (GB) units of sound speed $c_s = \sqrt{T_e/m_i}$ and the corresponding Larmor radius $\rho_s =$

c_s/ω_{ci}, where $\omega_{ci} = eB_{\rm ref}/m_i$ is the ion cyclotron frequency, and the index i (e) corresponds to the ions (electrons). The user should note that the definition of B used to define c_s and ρ_s may vary between codes and publications.

In the usual case where $m_i = m_{\rm ref}$, the sound speed is $c_s = v_{\rm thref}\sqrt{T_{Re}/2}$, the Larmor radius is $\rho_s = \rho_{\rm ref}\sqrt{T_{Re}/2}$, and the GKW wavevectors have $k_\perp \rho_{\rm ref} = k_\perp \rho_s \sqrt{T_{Re}/2}$. [1]

The time normalisation converts to GB units as

$$t_N = \frac{v_{\rm thref}}{R_{\rm ref}} t = \sqrt{\frac{2}{T_{Re}}} \frac{c_s}{R_{\rm ref}} = \frac{\sqrt{\frac{2}{T_{Re}}}a}{R_{\rm ref}} \left(\frac{c_s}{a}\right) t, \tag{10.23}$$

and the heat fluxes convert to a gyro-Bohm diffusivity as

$$\chi^N = \frac{\mathcal{I}_2}{R_{\rm ref}/L_T \{g^{\psi\psi N}\}} = \frac{R_{\rm ref}}{\rho_{\rm ref}^2 v_{\rm thref}} \chi = \left(\frac{T_{Re}}{2}\right)^{3/2} \frac{R_{\rm ref}}{a} \left(\frac{a}{\rho_s^2 c_s}\right) \chi. \tag{10.24}$$

Note that N in the two previous expressions is used to denote a normalised quantity, as used in most of this manual, and the equivalent quantity without the N is a dimensional one. The dimensional diffusivity in m^2/s is given by

$$\chi = \frac{\mathcal{I}_2 \rho_*^2 v_{\rm thref} R_{\rm ref}}{\{g^{\psi\psi N}\} R_{\rm ref}/L_T} \tag{10.25}$$

and a dimensional power flux in J/s is obtained (see A.7) by

$$Q = \{\mathbf{\Gamma}_2 \cdot \nabla\psi\} V' \tag{10.26}$$

where V is the dimensional volume of the plasma with the local flux surface (see A.6) and

$$V' = \frac{\partial V}{\partial \epsilon} = R_{\rm ref}^3 J^N_{\psi\zeta s}. \tag{10.27}$$

In the flux tube, one should realize that the *local* radial coordinate x_r appearing as the scale for the XY diagnostics (in file `xphi`) differs by a factor of ρ_* from the *global* radial coordinate $\psi = \epsilon$. Thus $\Delta x_r = \Delta\psi/\rho_* = \Delta r/\rho_{\rm ref}$. This is required when combining background and perturbed quantities in the flux tube to create a profile as in Sec. 5.4.

10.6 Output grids (module diagnos_grid)

The ζ grid output is re-scaled to the input scale of $k_\theta \rho_{\rm ref}$, so the two perpendicular coordinate scales are equal (Eq. 2.256).

10.7 Summary: diagnostic outputs and their data format

In GKW, most diagnostics are calculated inside the code at runtime (as opposed to in post-processing). The advantage of this approach is that one does not need to store the full 5 dimensional distribution functions at every timestep (which occupy a lot of disk space and are difficult to transfer). The disadvantage is that if a time dependant diagnostic is not enabled or implemented before the run, that information cannot later be retrieved without re-running the entire simulation (final state diagnostics can always be retrieved by restarting the code if the restart file is available). One should therefore think carefully about which diagnostics are required before launching any expensive simulation. One could argue that this approach is suited to a world in which computation is cheap, but storage and transfer are expensive.

[1] Care should also be taken when comparing the GKW input wavenumber "k_θ" with other codes. It is defined by Eq. 2.256, and the relationship to the toroidal mode number is described by Eq. 2.257.

The default set of time dependent diagnostics contains scalar time traces (fluxes, growth rates) and 1D spectra, but no 2D or 3D quantities.

GKW does contain a wrapper which allows to write output data in various formats, according to the setting of *io_format* in the input file.

io_format='ascii' (also called 'formatted' output) All data is written to a collection of text files. Those can be read and analysed easily, and they are very portable. The ASCII format becomes impractical for higher dimensional and large amounts of data.

For ascii and binary formats, diagnostic meta-data is output by the code to a simple text file `gkwdata.meta`.

io_format='binary' (also called 'raw' output) All data is written to binary files. Binary data consumes typically by a factor of about 6-10 less disk space than ascii data. Binary or hdf5 (see below) format is therefore recommended if large amounts of data are produced.

io_format='hdf5' (also called 'structured' output), if *io_format='hdf5'* is set. This structured binary format is available if the code was compiled with the HDF5 library. If the HDF5 library was compiled with szip encoding support, then the GKW output data can be compressed (for now, there is only a hardcoded switch, see the comments).
The HDF5 format allows to output diagnostic meta-data by associating attributes to datasets. These can be read by analysis scripts. More and more metadata will be added in the near future, for discussions see issue 69. Note that

- If one tries to access HDF5 files during run, they may turn out to be invalid (not always though).
- Aborted runs may leave a corrupted HDF5 file. Usually then there are just no parameters in the file, sometimes datasets are missing and in rarer cases even the file as a whole is invalid.
- To produce the 3d fields and moments data and the 4d mode structure data, a MPI gather operation is performed and the output is written serially by just one process. For large runs with many processes this is likely not very performant. Instead, it is recommended to use MPI-IO binary output format then.
- At the time moment GKW supports only serial HDF5.

In order to take advantage of the benefits of certain formats, GKW knows also about mixed output:

io_format='mixed' Most of the data (mainly scalar time traces and 1D spectra, `parallel.dat` final state eigenfunctions) is output to text files, whereas higher dimensional data is output to binary files.

io_format='hdf5+ascii' All data is output as with the 'hdf5' setting. In addition, lowerdimensional data is output also to text files.

As long as there is not enough descriptive metadata attached to the GKW diagnostic output data, we here summarize most of the GKW diagnostic outputs in tables 10.1 and 10.2 for 1D time traces, final state outputs, and 2D time trace outputs. A full listing of available diagnostics and switches is documented in input.dat.sample. In the case of less standard diagnostics it may be necessary to examine the source and comments in the respective diagnostic module diagnos_foo.F90 to determine the exact order and meaning of the output quantities.

Generally, 2D diagnostics can be output by GKW at every timestep, or at the end ("R or E" in tables 10.1 and 10.2, switched by *xy_estep*).
When successive binary records for each timestep are stacked into one file, this happens with (32 bit / 4 byte, usually) integers before and after each record for the size of the binary record (which will have Endianness of the hardware it was written on). The data records can be read into matlab with the script read_file2d.m.

Some slices are output at the point nearest the LFS, for these a different location can be set with switch *xy_slice_ipar*.

Reorganisation of diagnostics and the *io_legacy* switch: Note that there currently efforts to clean up GKW's diagnostics. This includes renaming of some files, split up of datasets and change of the order of the data, where necessary. The goal is to output data in a more natural way which is easy to understand and to analyse.

In order to provide users for some time the possibility to keep using their old analysis tools without larger changes, the switch *io_legacy* was introduced.

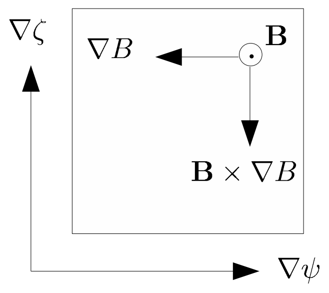


Figure 10.1: Coordinate directions on the outboard plane ($s = 0$) for $s_{\rm B} = 1$, as output by the potential slices diagnostic `phi#` with the coordinate labels for each point as output by `xphi` and `yphi` (the yphi coordinate label inverts the plot). For $s_{\rm B} = 1$, the poloidal field is 'upwards' $\mathrm{sign}(B_p \nabla\theta \cdot \nabla\zeta) = s_{\rm B}$. If in doubt, see Sec. 2.2.6. Note that the 'centre' of the flux tube is in the bottom left corner when output from the FFTs.

Filename / dataset name	Description	Quantity (norm.=normalised)	Size	when	Grid Files	Namelist Switch
		Module diagnos_growth_freq:				
`dominant_growth_rate`	Dominant linear mode growth rate	$\gamma_N^{\max}$	N_t	R		
`dominant_real_freq`	Dominant linear mode frequency	ω_N belonging to $\gamma_N^{\max}$	N_t	R		
`growth_rates.dat`	Growth rates of all linear modes	γ_N		R		
`frequencies.dat`	Frequencies of all linear modes	ω_N		R		
`amplitudes.dat`	Amplitudes of all linear modes			R		
		Module diagnos_grid:				
`time.dat`	timestamp (large time steps)	t_N	N_t	R		
`sgrid`	s grid (parallel coordinate)	s	N_s	B		
`krho`	Bi-normal wavevector grid	$(q/2\pi\psi)k_\zeta\rho_{\rm ref} = k_\perp(k_\psi = 0, s = 0)\rho_{\rm ref}$	$N_x, N_{\rm mod}$	B		
`kxrh`	Radial wavevector grid	$k_\psi\rho_{\rm ref}$	$N_x, N_{\rm mod}$	B		
`yphi`	Real space bi-normal grid	$2\pi\psi(\zeta - \zeta_{\rm min})/q\rho_{\rm ref} = k_\theta\rho_{\rm ref}$	$M_x, M_{\rm mod}$	B		
`xphi`	Real space radial grid	$(\psi - \psi_{\rm min})/\rho_* = (r - r_{\rm min})/\rho_{\rm ref}$	$M_x, M_{\rm mod}$	B		
`kx_connect`	Parallel boundary connections	Labelled in file	N/A	B		
		Module diagnos_mode_struct:				
`phi`	es. potential	See Sec. 10.7.1.	$N_{\rm MOD}, N_s, N_x$	E and O		
`Apar`		See Sec. 10.7.1.	$N_{\rm MOD}, N_s, N_x$	E and O		
`Bpar`		See Sec. 10.7.1.	$N_{\rm MOD}, N_s, N_x$	E and O		
`dens`	density fluctuations	See Sec. 10.7.1.	$N_{\rm sp}, N_{\rm MOD}, N_s, N_x$	E and O		
`Tpar`	parall. temperature fluct.	See Sec. 10.7.1.	$N_{\rm sp}, N_{\rm MOD}, N_s, N_x$	E and O		
`Tperp`	perp. temp. fluctuations	See Sec. 10.7.1.	$N_{\rm sp}, N_{\rm MOD}, N_s, N_x$	E and O		
`vpar`	parallel flow fluctuations	See Sec. 10.7.1.	$N_{\rm sp}, N_{\rm MOD}, N_s, N_x$	E and O		
		Module diagnos_fields:				
`kyspec`	Bi-normal mode potential spectrum	$\sum_{k_\psi}\int_s \vert\hat\phi(k_\psi, k_\zeta, s)\vert^2\,{\rm d}s$	$N_t, N_{\rm mod}$	R		
`kyspec_em`	(Equivalent for $A_\Vert$ field)	$\sum_{k_\psi}\int_s \vert\hat A_\Vert(k_\psi, k_\zeta, s)\vert^2\,{\rm d}s$	$N_t, N_{\rm mod}$	R		
`kxspec(_em)`	Radial mode potential spectrum	$\sum_{k_\zeta}\int_s \vert\hat\phi(k_\psi, k_\zeta, s)\vert^2{\rm d}s$	N_t, N_x	R		
`kyvort`	Bi-normal vorticity spectra	$\sum_{k_\psi,sp}\int Z_{sp}\vert\hat f_{sp}(k_\psi, k_\zeta, s)\vert^2\,{\rm d}^3v\,{\rm d}s$	$N_t, N_{\rm mod}$	R		
`kxvort`	Radial vorticity spectra	$\sum_{k_\zeta,sp}\int Z_{sp}\vert\hat f_{sp}(k_\psi, k_\zeta, s)\vert^2\,{\rm d}^3v\,{\rm d}s$	N_t, N_x	R		
`parallel_phi`	ES potential parallel structure (no ZF)	$\sum_{k_\zeta\neq 0}\sum_{k_\psi}\hat\phi(k_\psi, k_\zeta, s)$	N_t, N_s	R	`time,sgrid`	*lparallel_phi*
`parallel_apar`	EM potential parallel structure (no ZF)	$\sum_{k_\zeta\neq 0}\sum_{k_\psi}\hat A_\Vert(k_\psi, k_\zeta, s)$	N_t, N_s	R	`time,sgrid`	*lparallel_apar*
`parallel_bpar`	EM field parallel structure (no ZF)	$\sum_{k_\zeta\neq 0}\sum_{k_\psi}\hat B_\Vert(k_\psi, k_\zeta, s)$	N_t, N_s	R	`time,sgrid`	*lparallel_bpar*
`spc#`	ES potential spectrum (LFS)	$\vert\hat\phi(k_\psi, k_\zeta, s \approx 0)\vert$ (norm.)		R or E	`kxrh, krho`	*xy_phi*
`phi#`	ES potential slice (LFS)	$\phi(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	*xy_phi*
`apc#`	EM potential spectrum (LFS)	$\vert\hat A_\Vert(k_\psi, k_\zeta, s \approx 0)\vert$ (norm.)		R or E	`kxrh, krho`	*xy_apar*
`apa#`	EM potential slice (LFS)	$A_\Vert(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	*xy_apar*
`bpc#`	EM field spectrum (LFS)	$\vert\hat B_\Vert(k_\psi, k_\zeta, s \approx 0)\vert$ (norm.)		R or E	`kxrh, krho`	*xy_bpar*
`bpa#`	EM field slice (LFS)	$B_\Vert(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	*xy_bpar*
`vok#`	Vorticity spectrum (FSA)	$\sum_s\int \vert\hat f_s(k_\psi, k_\zeta)\vert d^3v ds$ (norm.)		R or E	`kxrh, krho`	*xy_vort*
		Module diagnos_moments:				
`ene[sp#]_#`	Temp. pert. moment (LFS)	$\sum_{sp}\int \vert\hat f_s v^2\vert d^3v$ (norm.)		R or E	`xphi, yphi`	*xy_temp*
`den[sp#]_#`	Density pert. moment (LFS)	$\int \vert\hat f_s\vert d^3v$ (norm.)		R or E	`xphi, yphi`	*xy_dens*
`pac[sp#]_#`	Current pert. moment (LFS)	$\int \vert\hat f_s\vert v_\Vert d^3v$ (norm.)		R or E	`xphi, yphi`	*xy_current*
`pac[sp#]_#`	Current^2 pert. moment (LFS)	$\int \vert\hat f_s\vert v_\Vert^2 d^3v$ (norm.)		R or E	`xphi, yphi`	*xy_current2*
`den/ene_spectra`	Bi-normal spectral perturbation moments for each species	$\int_s\int \hat\alpha_{1,2}\vert\hat f_{sp}\vert{\rm d}^3v\,{\rm d}s$	$N_{\rm mod}, N_{sp}$			

Table 10.1: Diagnostics output by GKW (Part 1), with *io_legacy=false*. The abbreviations denote output at the beginning (B), repeatedly (R), at the end (E) or other (O). FSA denotes Flux Surface Averaged quantities. LFS denotes quantities evaluated at the low field side, i.e. in the center of the s-grid.

Filename / dataset name	Description	Quantity (norm.=normalised)	Size	beginning (B), repeatedly (R), at the end (E)	Grid Files	Namelist Switch
		Module diagnos_fluxes:				
`p/e/vflux_es`	Total v_E particle flux by species	$\mathcal{I}_i = \sum_{k_\zeta} \sum_{k_\psi} \int_s \bar{\mathcal{I}}_i \, \mathrm{d}s$	N_t, N_{sp}	R	`time`	
`p/e/vflux_apar`	Total $A_\parallel$ particle flux by species	$\mathcal{J}_i = \sum_{k_\zeta} \sum_{k_\psi} \int_s \bar{\mathcal{J}}_i \, \mathrm{d}s$	N_t, N_{sp}	R	`time`	
`p/e/vflux_bpar`	Total $\nabla B_\parallel$ particle flux by species	$\mathcal{K}_i = \sum_{k_\zeta} \sum_{k_\psi} \int_s \bar{\mathcal{K}}_i \, \mathrm{d}s$	N_t, N_{sp}	R	`time`	
`p/e/vflux_spectra`	Bi-normal spectral v_E flux for each species	$\sum_{k_\psi} \int_s \bar{\mathcal{I}}_i(k_\psi, k_\zeta) \, \mathrm{d}s$	$N_t, N_{\mathrm{mod}}, N_{sp}$	R		
`p/e/vflux_xspec`	Radial spectral v_E fluxes for each species	$\sum_{k_\zeta} \int_s \mathcal{I}_i(k_\psi, k_\zeta) \, \mathrm{d}s$	$N_t, N_{\mathrm{mod}}, N_{sp}$	R		
`p/eflux_(em_)sup`	Bi-normal spectral flux suprema for each species	$\hat{\mathcal{I}}_i(k_\zeta), \hat{\mathcal{J}}_i(k_\zeta)$	$N_t, N_{\mathrm{mod}}, N_{sp}$	R		
`PFlesr[sp#]_#`	ES radial particle flux (LFS)	$\mathcal{I}_1(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	*xy_fluxes_P*
`EFlesr[sp#]_#`	ES radial energy flux (LFS)	$\mathcal{I}_2(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	*xy_fluxes*
`VFlesr[sp#]_#`	ES radial momentum flux (LFS)	$\mathcal{I}_3(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	*xy_fluxes_V*
`[P/E/V]Flemr[sp#]_#`	EM flutter fluxes as above (LFS)	$J_i(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	+ *xy_fluxes_em*
`[P/E/V]Flbpr[sp#]_#`	EM comp. fluxes as above (LFS)	$K_i(\psi, \zeta, s \approx 0)$ (norm.)		R or E	`xphi, yphi`	+ *xy_fluxes_bpar*
`[P/E/V]FS[es/em/bp]r[sp#]_#`	All fluxes as abv., but (FSA)	$\int I_i/J_i/K_i(\psi, \zeta) ds$ (norm.)		R or E	`xphi, yphi`	+ *xy_fluxes_fsa*
`[P/E/V]F[l/S][es/em/bp]p[sp#]_#`	Binormal fluxes as abv. (LFS/FSA)	N/A		R or E	`xphi, yphi`	+ *xy_fluxes_bi*
`[P/E/V]Fl[es/em/bp]k[sp#]_#`	Radial fluxes spectra (FSA)	$\int I_i/J_i/K_i(k_\psi, k_\zeta) ds$ (norm.)		R or E	`kxrh, krho`	+ *xy_fluxes_k*
		Module diagnos_fluxes_vspace:				
`fluxes_det.dat`	5D fluxes for each species, including d^3X and d^3v	$\left(\bar{\mathcal{I}}_i + \bar{\mathcal{J}}_i + \bar{\mathcal{K}}_i\right)(k_\psi, k_\zeta, s, \mu, v_\parallel)$	binary, $N_{v_\parallel} N_\mu N_s$ $N_\psi N_{mod} N_{sp} i$, $i = 1, 2, 3$	E		*lfluxes_detail*
`[p/e/v]fluxes_vspace[sp#]_#`	Fluxes in velocity space (FSA)	$\int I_i/J_i/K_i(v_\parallel, \mu) ds$ (norm.)		R or E	`distr[1/2]`	*lfluxes_vspace*
		Module diagnos_f:				
`distr*.dat`	2D velocity space output of 1 mode.	1,2 grids, 3 Re($\hat{g}(s=0)$), 4 Im($\hat{g}(s=0)$)	N_μ, $N_{v_\parallel}$	E		
		Output by other parts of the code:				
`parfun.dat`	Perpendicular wavevector	$k_\perp \rho_{\mathrm{ref}}$	$2, N_s N_{\mathrm{mod}} N_x$	B		
`par.dat`	Curvature and Coriolis functions	$k_\perp \rho_{\mathrm{ref}}$, $\mathcal{D}^\psi k_\psi + \mathcal{D}^\zeta k_\zeta$, $\mathcal{H}^\psi k_\psi + \mathcal{H}^\zeta k_\zeta$,	$3, N_s N_{\mathrm{mod}} N_x$	B		
`geom.dat`	Geometry tensors	Labelled in file	N/A	B		
`FDS`	Restart distribution function	binary (see Sec. 9.3.8)	E			
`FDS.dat`	Additional restart information	Ascii namelists (see Sec. 9.3.8)	E			
`Coll_params.dat`	Collision frequencies	Labelled in file	N/A	B		
`cfdens.dat`	Centrifugal poloidal density variations	s, $R/L^E_{\mathrm{n,sp}}(s)$, $exp(-\mathcal{E}_R)_{\mathrm{sp}}(s), \theta$	$2N_{\mathrm{sp}} + 2$	B		

Table 10.2: Diagnostics output by GKW (Part 2) , with *io_legacy=false*.

Filename	Description	Quantity (norm.=normalised)	Cols,(Rows)	beginning (B), repeatedly (R), at the end (E)	Grid Files	Namelist Switch
Module diagnos_growth_freq:						
`time.dat`	Dominant linear mode growth rate	$t_N, \gamma_N^{\max}, \omega_N$	3	R		
`growth.dat`	Growth rates of all linear modes	γ_N		R		
Module diagnos_mode_struct:						
`parallel.dat`	Parallel mode structure	See Sec. 10.7.1.	$N_{\mathrm{MOD}} N_x N_s N_{\mathrm{sp}}$	E		
Module diagnos_moments:						
`den/ene_spectra`	Bi-normal spectral perturbation moments for each species	$\int_s \int \hat{\alpha}_{1,2} \lvert \hat{f}_{sp} \rvert \mathrm{d}^3 v \, \mathrm{d}s$	$N_{\mathrm{mod}} N_{sp}$			
Module diagnos_fluxes:						
`fluxes.dat`	Total v_E fluxes $(i = 1, 2, 3)$ by species	$\mathcal{I}_i = \sum_{k_\zeta} \sum_{k_\psi} \int_s \bar{\mathcal{I}}_i \, \mathrm{d}s$	$3N_{sp}$	R		
`fluxes_em.dat`	Total δB fluxes $(i = 1, 2, 3)$ by species	$\mathcal{J}_i = \sum_{k_\zeta} \sum_{k_\psi} \int_s \bar{\mathcal{J}}_i \, \mathrm{d}s$	$3N_{sp}$	R		
`fluxes_bpar.dat`	Total $\nabla B_{1\parallel}$ fluxes $(i = 1, 2, 3)$ by species	$\mathcal{K}_i = \sum_{k_\zeta} \sum_{k_\psi} \int_s \bar{\mathcal{K}}_i \, \mathrm{d}s$	$3N_{sp}$	R		

Table 10.3: Legacy diagnostics output by GKW, with *io_legacy=true*. This table shows output datasets which differ from the ones described in tables 10.1 and 10.2 which are associated with *io_legacy=false*.

10.7.1 Mode structure (module `diagnos_mode_struct`)

The file `parallel.dat` contains some diagnostics connected with the parallel mode structure. For every mode then species the following quantities are (sequentially) written to file, in this column order:

```
1     sgr         the length along the field line
2-3   phi         the perturbed potential
4-5   apar        the parallel component of the vector potential
6-7   dens        the perturbed density
8-9   tpar        the perturbed parallel temperature
10-11 tperp       the perturbed perpendicular temperature
12-13 wflow       the perturbed parallel flow velocity
14-15 bpar        the perturbed parallel (compressional) magnetic field
```

Note only *sgr* is real. The other quantities are complex and fill 2 columns in the output file. For a normalised run, all quantities are normalised and rotated in the complex plane so that the maximum of the potential for each mode is $1 + 0i$. Therefore the output quantity is $\hat{l}_E(s)$ where

$$\hat{l}(s,t) = \exp(\gamma t + i\omega t)\hat{l}_E(s). \tag{10.28}$$

The file will have $N_{\rm MOD} N_x N_s N_{\rm sp}$ rows (with data ordered by ζ mode, then ψ mode, then species). For a run with multiple modes, extracting data of interest can be a little involved, matlab script parallelldat.m, helps with this.

10.8 Island torque and stabilization by the parallel current

A diagnostic especially useful for tearing mode dynamics is the one giving the torque and the stabilization by the parallel current, namely T_φ and $\Delta_{\rm pol}$, respectively. These quantities are defined as follows, under the assumption of a non rotating island

$$T_\varphi \propto \int_{-L_x/2}^{L_x/2} dx \int_{-\pi}^{\pi} d\zeta_0 \; J_\parallel \sin\left(\kappa_{\zeta,\rm isl}\zeta_0\right) \tag{10.29}$$

$$\Delta_{\rm pol} \propto \int_{-L_x/2}^{L_x/2} dx \int_{-\pi}^{\pi} d\zeta_0 \; J_\parallel \cos\left(\kappa_{\zeta,\rm isl}\zeta_0\right) \tag{10.30}$$

where L_x is the width of the simulation box in the radial direction, the variable ζ_0 is the helical co-ordinate in the absence of rotation and $J_\parallel$ is the gyro-centre parallel current, given by

$$J_\parallel = \sum_s eZ_s \int_{\mathbb{R}^3} d^3\mathbf{v}\, v_\parallel \delta f_s \tag{10.31}$$

with δf_s the perturbed distribution function of the gyro-centres of species s. Note that multiplying equation 10.29 by $-i$ and adding the resulting expression to equation 10.30 yields

$$\Delta_{\rm pol} - iT_\varphi \propto \sum_s eZ_s \int_{-L_x/2}^{L_x/2} dx \int_{-\pi}^{\pi} d\zeta_0 \int_{\mathbb{R}^3} d^3\mathbf{v}\, v_\parallel \delta f_s \, \exp\left(-i\kappa_{\zeta,\rm isl}\zeta_0\right) \tag{10.32}$$

where we can distinguish the Fourier mode $\delta\hat{f}_{s\,isl} \equiv \delta\hat{f}_s\left(\kappa_\zeta = \kappa_{\zeta,\rm isl}\right) \propto \int_{-\pi}^{\pi} d\zeta_0\, \delta f_s \, \exp\left(-i\kappa_{\zeta,\rm isl}\zeta_0\right)$, with $\kappa_{\zeta,\rm isl}$ being the mode of the island in the binormal direction. Therefore, we can write

$$\Delta_{\rm pol} - iT_\varphi \propto \sum_s eZ_s \int_{-L_x/2}^{L_x/2} dx \int_{\mathbb{R}^3} d^3\mathbf{v}\, v_\parallel \delta\hat{f}_{s\,\rm isl} \tag{10.33}$$

A rotating island is usually imposed or obtained self-consistently in GKW. This means that the island is not always in the middle of the simulation box in the binormal direction. However, the above expressions are obtained assuming that the island is centered and therefore need to be corrected by multiplying by the phase $\exp\left(i\int_0^t dt'\omega_{\rm isl}\right)$, where $\omega_{\rm isl}$ is the rotation of the island, which results in the definition of the co-ordinate $\zeta = \zeta_0 - \int_0^t dt'\omega_{\rm isl}$. Note that in the absence of rotation, the amplitude of the vector potential $A_\parallel$ is real, due to the fact that a pure $\cos\left(\kappa_\zeta\zeta\right)$ is represented in GKW by a real amplitude of $A_\parallel$. Therefore, any departure from a real amplitude in $A_\parallel$ implies a rotation $\omega_{\rm isl}$, which can be calculated as

$$\int_0^t dt'\omega_{\rm isl} = \arctan\left(\frac{\Im\left(A_\parallel\right)}{\Re\left(A_\parallel\right)}\right) \tag{10.34}$$

leading to the expression

$$\mathcal{T} \equiv \Delta_{\rm pol} - iT_\varphi \propto \sum_s eZ_s \int_{-L_x/2}^{L_x/2} dx \int_{\mathbb{R}^3} d^3\mathbf{v}\, v_\parallel \delta\hat{f}_{s\,\rm isl} \exp\left(i\int_0^t dt'\omega_{\rm isl}\right) \tag{10.35}$$

Finally, the torque and instability parameter are calculated as

$$T_\varphi = -\Im\left(\mathcal{T}\right) \tag{10.36}$$

$$\Delta_{\rm pol} = \Re\left(\mathcal{T}\right) \tag{10.37}$$

These two quantities are output in the files `torque.dat` and `deltaprime.dat`, respectively. The first column corresponds to the values for the ions and the second column to the values for the electrons.

10.9 Entropy (module `diagnos_energetics` and `diagnos_eng`)

10.9.1 Definition of the Total Entropy of the Simulated Box

Let $p(\mathbf{X}, v_\|, \mu) \in [0,1]$ be a probability density function.

One can define the functional

$$S = -\int \mathrm{d}^3x\mathrm{d}^3v p \ln(p \cdot c_d), \qquad c_d = \text{const.} \tag{10.38}$$

and show that this expression has properties of the thermodynamical entropy (see e.g. the book of Jelitto 1989). It is therefore called "statistical entropy" or Shannon-Jaynes-entropy or Pauli-entropy or Boltzmann-Shannon-Gibbs (BGS) entropy. A constant c_d is formally needed in 10.38 to make the logarithm dimensionless [77] but it can be set equal to 1. Note that the integral in 10.38 is over the whole system, i.e. derived integrals below, on quantities of the simulated model, are taken over the whole simulation box.

A "relative entropy" functional can be defined in the form

$$S = -\int \mathrm{d}^3x\mathrm{d}^3v \; p \ln \frac{p}{q} \tag{10.39}$$

This functional characterizes the distribution p with respect to a certain reference probability density function $q \in [0,1]$.

Maximization of the relative entropy S under the constraint

$$1 = \int \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v} \, p(\mathbf{x}, \mathbf{v}) \tag{10.40}$$

leads to the condition

$$0 = 1 + ln\frac{p}{q} + \alpha \tag{10.41}$$

and thus

$$p = q \cdot \exp[-(1+\alpha)] \tag{10.42}$$

the distribution which corresponds to the extremum of the functional depends on the reference. With the assumption that the reference PDF q fulfills 10.40, one applies this constraint 10.40 to 10.42 and finds then $\alpha = -1$ and so

$$p = q, \tag{10.43}$$

that means the entropy has its extremum for $p = q$. Variation of the relative entropy under other constraints, using a reference which fulfills those, leads to the condition $p = q$ by the same argument.

For a PDF describing an ensemble of classical particles, the distribution which maximizes the entropy under two contraints (normalisation of the distribution and given kinetic energy average) is the Maxwellian. For our purposes, we put therefore f_{tot} in place of p and the reference distribution is chosen to be the Maxwell distribution in velocity space and to describe particle trapping (and possibly centrifugal trapping) in position space. This Maxwellian distribution function is denoted with F_M and is given in equation 1.29.

$$S = -\int \mathrm{d}^3X\mathrm{d}^3v f_{tot} \ln \frac{f_{tot}}{F_M} \tag{10.44}$$

A state of maximum entropy then corresponds to $f_{tot} = F_M$. In the δf model, where the distribution is split into $f_{tot} = F_M + f$, this means that the distribution perturbation vanishes in the thermodynamic equilibrium: $f = 0$.

Furthermore, it can be shown [77] that in contrast to the BGS entropy 10.38 given first, maximizing the relative entropy 10.39 leads to the same distribution function, independently of the coordinate system. This is an important property if we want to investigate systems relaxing towards equilibrium in the field aligned coordinate system in which the gyrokinetic model is formulated. The invariance under coordinate transformations of the maximum-entropy principle associated with the relative entropy notion 10.39 guarantees that the distribution $f_{tot} = F_M$ really corresponds to the maximum value of the entropy, and not another distribution, e.g. by multiplication with a factor.

Reduction of the perturbation $|f|$ means growth of the entropy S, i.e. the system moves towards thermodynamic equilibrium. On the contrary, processes that grow fluctuations $|f|$ effectively diminish the entropy of the system and keep it away from equilibrium, which is by definition the macrostate of maximum entropy.

We set $f_{tot} = F_M + f$ and expand around $f = 0$:

$$\begin{aligned}(F_M + f)\ln\frac{F_M + f}{F_M} &= (F_M + f)\left[0 + \frac{1}{F_M + f}|_{f=0}f^1 + \frac{1}{2}\frac{-1}{(F_M + f)^2}|_{f=0}f^2 + \dots\right]\\ &= 0 + f + \frac{f^2}{F_M} - \frac{1}{2}\frac{f^2}{F_M} + \dots\\ &= 0 + f + \frac{1}{2}\frac{f^2}{F_M} + \dots \qquad (10.45)\end{aligned}$$

Then 10.44 becomes

$$S \approx -\int \mathrm{d}^3X\mathrm{d}^3v(f + \frac{1}{2}\frac{f^2}{F_M}) = -\frac{1}{2}\int \mathrm{d}^3X\mathrm{d}^3v\frac{f^2}{F_M} \qquad (10.46)$$

where the integral of the fluctuation f vanishes. S in 10.46 is a small-amplitude approximation to the plasma entropy. Again, one can see that reduction of $|f|$ means increase of the entropy S.

In its normalized form 10.46 is

$$S \approx -\frac{1}{2}\int \mathrm{d}^3X\mathrm{d}^3v\frac{f^2}{F_M} = \rho_*^2\frac{v_{thref}}{R_{ref}}n_{R_0}\underbrace{\left(-\frac{1}{2}\int \mathrm{d}^3X\mathrm{d}^3v_N\frac{f_N^2}{F_{MN}}\right)}_{S_N} \qquad (10.47)$$

In the normalized balance equation 10.49 the reference quantities are dropped, but the smallness parameter ρ_* is kept for illustration.

The quantity $H = -S$ is a measure of the intensity of the fluctuations in the distribution.

$$H \approx \frac{1}{2}\int \mathrm{d}^3X\mathrm{d}^3v\frac{f^2}{F_M} = \rho_*^2\frac{v_{thref}}{R_{ref}}n_{R_0}\underbrace{\left(\frac{1}{2}\int \mathrm{d}^3X\mathrm{d}^3v_N\frac{f_N^2}{F_{MN}}\right)}_{H_N} \qquad (10.48)$$

To emphasise, the fluctuation intensity H is found to be *increasing* in the simulations of a system driven out of equilibrium by applied thermodynamic forces (temperature and density gradient). Accordingly the entropy $S = -H$ is found to decrease.

10.9.2 Entropy Balance

General procedure

One starts from the normalized gyrokinetic equation given in section 2.4, multiplies every term with $\frac{f}{F_M}$ and integrates over the box volume and velocity space.

In this section, all quantities are normalized, but we omit the index N here. Moreover, only the electrostatic case is considered, so that $g = f$ for the purposes of this section.

$$\rho_*^2\int \mathrm{d}^3X\mathrm{d}^3v\frac{f}{F_M}\left[\frac{\partial g}{\partial t} + \mathrm{I} + \mathrm{II} + \mathrm{III} + \mathrm{IV} + \mathrm{V} + \mathrm{VI} + \mathrm{VII} + \mathrm{VIII}\right] = 0 \qquad (10.49)$$

Regard the terms separately

One can recover $\frac{\partial}{\partial t}H$ in the very first integral.

$$\begin{aligned}\rho_*^2 \int \mathrm{d}^3X\mathrm{d}^3v \frac{f}{F_M}\frac{\partial g}{\partial t} &= \rho_*^2 \int \mathrm{d}^3X\mathrm{d}^3v \frac{1}{F_M}\frac{1}{2}\frac{\partial f^2}{\partial t} \\ &= \frac{\partial H}{\partial t} = -\frac{\partial S}{\partial t}\end{aligned} \tag{10.50}$$

The linear $\mathbf{E}\times\mathbf{B}$-drift term V yields

$$\begin{aligned}\rho_*^2 \int \mathrm{d}^3X\mathrm{d}^3v \frac{f}{F_M}\ \mathrm{V} = \rho_*^3 \int \mathrm{d}^3X \Bigg[&-\frac{1}{L_n}\Gamma_N^\psi - Q_N^\psi \frac{1}{L_T} \\ &+ \frac{3}{2}\frac{1}{L_T}\Gamma^\psi - \frac{\mathcal{E}_R}{T_R}\frac{1}{L_T}\Gamma^\psi + \\ &- \frac{2}{T_R}\Pi_\varphi^\psi u' - 2\frac{m_R}{T_R}\Omega_N^2 \mathcal{L}\,\Gamma^\psi \Bigg]\end{aligned} \tag{10.51}$$

At the moment these six terms are output seperately by the code, but very likely they will be combined in the near future to fewer, more symmetric quantities.

The Landau-Damping terms VI and VII give non-vanishing contributions, too. In combination with the Poisson equation, they can be rewritten as

$$\rho_*^2 \int \mathrm{d}^3X\mathrm{d}^3v \frac{f}{F_M}\Big[\mathrm{VII} + \mathrm{VIII}\Big] = \int \mathrm{d}^3X \frac{\partial}{\partial t}\sum_s \int \mathrm{d}^3v \frac{Z_s^2}{2T_s^2}F_M\left(\phi^2(\mathbf{x}) - (\mathbf{G}\,\phi)^2\right) \tag{10.52}$$

$$\text{and } W_s \text{ is defined such that} = \frac{\partial}{\partial t}\sum_s W_s \tag{10.53}$$

$$W_{N,s} = \int \mathrm{d}^3X_N \int \mathrm{d}^3v_N \frac{Z_s^2}{2T_{R,s}^2}F_{MN}\left(\phi_N^2 - (\mathbf{G}\,\phi_N)^2\right) \tag{10.54}$$

If only ion dynamics is simulated and electrons are treated adiabatically, the index s above only runs over ion species and the term $W_{N,e}$ for the electrons is different.

$$\begin{aligned}\frac{\partial}{\partial t}W_{N,e,adia} &= \frac{\partial}{\partial t}\int \mathrm{d}^3X_N\phi_N \int \mathrm{d}^3v_N \frac{Z_e}{T_{R,e}}\left(\mathbf{G}^\dagger f_{eN}\right) \\ &= \frac{\partial}{\partial t}\int \mathrm{d}^3X_N\phi_N\frac{Z_e}{T_{R,e}}n_R\frac{Z_e}{T_{R,e}}\left(\phi_N - \{\phi_N\}_{\mathsf{FSA}}\right) \\ &= \int \mathrm{d}^3X_N n_R \frac{Z_e^2}{2T_{R,e}^2}\left(\frac{\partial\phi_N^2}{\partial t} - 2\phi_N\frac{\partial}{\partial t}\{\phi_N\}_{\mathsf{FSA}}\right)\end{aligned} \tag{10.55}$$

And with $\{.\}_{\mathsf{FSA}} = \{.\}_{\mathsf{FSA}}(\psi,\zeta)$ and $T_{R,e}^2$ being constant with respect to the parallel direction, one can write

$$= \int \mathrm{d}\psi\mathrm{d}\zeta J_X n_R \frac{Z_e^2}{2T_{R,e}^2}\left(\frac{\partial\{\phi_N^2\}_{\mathsf{FSA}}}{\partial t} - 2\{\phi_N\}_{\mathsf{FSA}}\frac{\partial}{\partial t}\{\phi_N\}_{\mathsf{FSA}}\right) \tag{10.56}$$

$$= \int \mathrm{d}\psi\mathrm{d}\zeta J_X n_R \frac{Z_e^2}{2T_{R,e}^2}\left(\frac{\partial\{\phi_N^2\}_{\mathsf{FSA}}}{\partial t} - \frac{\partial\{\phi_N\}_{\mathsf{FSA}}^2}{\partial t}\right) \tag{10.57}$$

$$= \frac{\partial}{\partial t}\int \mathrm{d}\psi\mathrm{d}\zeta J_X n_R \frac{Z_e^2}{2T_{R,e}^2}\left(\{\phi_N^2\}_{\mathsf{FSA}} - \{\phi_N\}_{\mathsf{FSA}}^2\right)\ . \tag{10.58}$$

10.9.3 Resulting entropy balance equation

The terms of the entropy balance equation which were very briefly presented in section 10.9.2, and the numerical diffusion terms, summed over all species make up the following balance equation.

$$
\begin{aligned}
0 = \sum_s \frac{\partial}{\partial t}(H_s + \overbrace{W_s}^{\text{from VII and VIII}}) + \sum_s \Big(& \overbrace{\text{contrib. from turb. fluxes}}^{\text{from linear ExB drift V;}} + \overbrace{\text{contrib. from neoclass. fluxes}}^{\text{from VI, but this is ignored and not implemented}} + \\
& - \mathcal{D}_{f\parallel} - \mathcal{D}_{v_\parallel} - \mathcal{D}_\perp \Big) - \sum_s \int \mathrm{d}^3X \mathrm{d}^3 v_N \frac{f_N}{F_{MN}} \mathcal{C}_N
\end{aligned}
\tag{10.59}
$$

To be more explicit, in terms of entropy this equation reads

$$
\begin{aligned}
\sum_s \frac{\partial}{\partial t}(S_s - W_s) = \sum_s \Big(\int \mathrm{d}^3X \Big[& -\frac{1}{L_n}\Gamma_N^\psi - Q_N^\psi \frac{1}{L_T} + \frac{3}{2}\frac{1}{L_T}\Gamma^\psi - \frac{\mathcal{E}_R}{T_R}\frac{1}{L_T}\Gamma^\psi - \frac{2}{T_R}\Pi_\varphi^\psi u' - 2\frac{m_R}{T_R}\Omega_N^2 \mathcal{L}\,\Gamma^\psi \Big] \\
& - \mathcal{D}_{f\parallel} - \mathcal{D}_{v_\parallel} - \mathcal{D}_\perp \Big) - \sum_s \int \mathrm{d}^3X \mathrm{d}^3 v_N \frac{f_N}{F_{MN}} \mathcal{C}_N
\end{aligned}
\tag{10.60}
$$

and in terms of free energy

$$
\begin{aligned}
& \frac{\partial}{\partial t} \sum_s (-T_{R,s} S_{N,s} + E_{\chi,N,s}) = \\
& \quad \sum_s T_{R,s} \Big(\int \mathrm{d}^3X \Big[\frac{1}{L_n}\Gamma_N^\psi + Q_N^\psi \frac{1}{L_T} - \frac{3}{2}\frac{1}{L_T}\Gamma^\psi + \frac{\mathcal{E}_R}{T_R}\frac{1}{L_T}\Gamma^\psi + \frac{2}{T_R}\Pi_\varphi^\psi u' + \frac{m_R}{T_R}\Omega_N^2 \mathcal{L}\,\Gamma^\psi \Big] \\
& \qquad + \mathcal{D}_{f\parallel} + \mathcal{D}_{v_\parallel} + \mathcal{D}_\perp \Big) + \sum_s \mathcal{D}_{C,s}
\end{aligned}
\tag{10.61}
$$

where $E_{\chi,S} = W_s T_{R,s}$.

In a system driven out of equilibrium the fluctuations grow, that is, the entropy gets smaller (i.e. more negative). The time derivative on the left side is therefore negative. This corresponds to the fluxes and the gradient lengths defined here being positive and the numerical dissipation terms $\mathcal{D}_i$ being negative.

In a statistically steady state, the left side of this equation vanishes. The entropy sinks from the fluxes are then balanced by the numerical dissipation terms.

10.9.4 Output quantities of the diagnostic

If `lcalc_energetics` in the diagnostics namelist is switched on, the code will produce a number of files related to the entropy balance equation:

The quantities in the entropy balance equation 10.60 correspond directly to the output of the `energetics()` diagnostic.

$$
\begin{aligned}
\texttt{dt\_entr} + \texttt{dt\_entr\_field} = -\Big(& \texttt{entr\_src01} + \texttt{entr\_src02} + \texttt{entr\_src03} \\
& + \texttt{entr\_src04} + \texttt{entr\_src05} + \texttt{entr\_src06} \\
& + \texttt{entr\_num\_dis} + \texttt{entr\_num\_vp} + \texttt{entr\_num\_perp} \\
& + \texttt{entr\_coll} \Big)
\end{aligned}
\tag{10.62}
$$

To clarify the signs, note the definition

$$
\frac{\partial}{\partial t} \sum_s (-W_{N,s}) = \texttt{dt\_entr\_field}\ . \tag{10.63}
$$

In order to check the balance one can sum both sides and see how well they eliminate each other.

$$\begin{aligned}&\texttt{dt\_entr} + \texttt{dt\_entr\_field} + \Big(\texttt{entr\_src01} + \texttt{entr\_src02} + \texttt{entr\_src03}\\&\quad + \texttt{entr\_src04} + \texttt{entr\_src05} + \texttt{entr\_src06}\\&\qquad + \texttt{entr\_num\_dis} + \texttt{entr\_num\_vp} + \texttt{entr\_num\_perp} + \texttt{entr\_coll}\Big) \overset{?!}{=} 0\end{aligned} \tag{10.64}$$

Chapter 11

Benchmarks

In this section we describe a set of benchmarks. The benchmark cases have been chosen to maximise the effect of each of the different implemented terms as much as possible. Together they give evidence for the correct implementation of the entire model. Linear benchmarks have been made against the GS2 code which is well established. Indeed the model equations are equivalent, with the exception that GKW includes the effect of a rigid body toroidal rotation, while GS2 does not. The implementation is to some extent similar; both use a spectral approach for the plane perpendicular to the magnetic field, and finite differences for the other directions. The main difference lies in the velocity space discretisation where GKW uses $(v_\parallel, \mu)$ coordinates, whereas GS2 uses pitch angle and energy. The GS2 velocity grid allows an efficient implicit time integrator, while the default grid of GKW is suitable for an explicit time integrator which facilitates parallelisation. Also the finite difference scheme used for GKW in the benchmarks is fourth order while GS2 has a second order scheme. In this section all quantities have been converted to use gyro-Bohm units.

11.1 Linear Benchmarks

The standard benchmark for linear problems is the growth rate of the Ion Temperature Gradient mode (ITG) as a function of the poloidal wave vector $k_\theta \rho_s$ for the so-called Cyclone base case [48] ($q = 1.4$, $\hat{s} = 0.78$, $\epsilon = 0.19$, $R/L_T = 6.9$, $R/L_n = 2.2$, $T_e/T_i = 1$, electro-static (zero $A_\parallel$), and adiabatic electrons). The growth rate for this case is shown in the right panel of Fig. 11.1 as a function of $k_\theta \rho_s$ for various values of the normalised ion temperature gradient ($R/L_T = 6.9$, 8.28 10.35, 12.44, and 15.18) calculated with both GKW and GS2. As can be seen from the figure, good agreement is obtained. The left panel, furthermore, shows the comparison of the potential as a function of the coordinate along the field line (s) for both codes.

For nonlinear runs the proper response of the zonal flows is of utmost importance. The standard benchmark [15, 69, 9] that addresses the physics of the zonal flow / geo-acoustic mode is the Rosenbluth-Hinton test which was described analytically in Ref. [72]. In this benchmark the initial condition is an ion density perturbation with a finite (small) radial wave vector, and no dependence on either s or ζ. The adiabatic electron response is used, keeping the correction due to the flux surface average of the potential. The density perturbation generates a potential perturbation and excites the so called geo-acoustic mode. This mode is damped and a small residual poloidal flow remains. Fig. 11.2 gives the potential perturbation as a function of time. The relevant parameters used for this benchmark are $q = 1.3$, $\epsilon = 0.05$, $k_\psi \rho_s = 0.02$, electrostatic, collisionless. Rather large grid sizes ($N_s = 128$ and $N_{v_\parallel} = 128$) are used to avoid the recurrence problem [15]. The residual potential is given by the equation

$$\frac{\phi(t = \infty)}{\phi(t = 0)} = \frac{1}{1 + q^2 \Theta/\epsilon^2} \qquad \text{with} \qquad \Theta = 1.6\epsilon^{3/2} + 0.5\epsilon^2 + 0.36\epsilon^{5/2} \tag{11.1}$$

The original derivation of Ref. [72] is accurate to lowest order in ϵ only, i.e. it retained only the first term ($1.6\epsilon^{3/2}$) in Θ. The higher order terms have been calculated in Ref. [73]. For the parameters of the simulation shown in the left panel the residue is $\phi(t = \infty)/\phi(t = 0) = 0.0713$ smaller than the result of Ref.

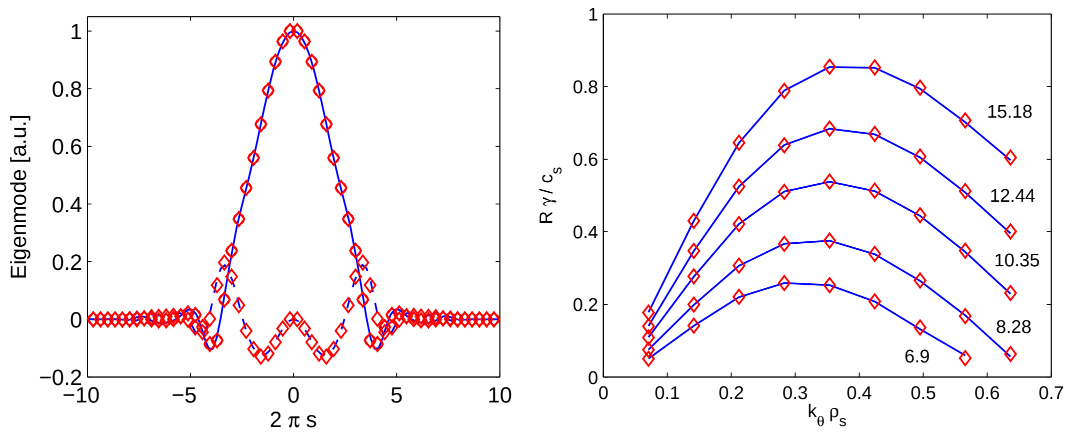


Figure 11.1: Left: mode structure of the Cyclone base case. The full line is the real part and the dotted line is the imaginary part of the potential calculated with GS2. The corresponding results of GKW are given by the diamonds. Right: growth rate as a function of $k_\theta \rho_s$ for various values of R/L_{Ti} (numbers given in the figure). The full lines give the results of GS2 while the diamonds are the corresponding results of GKW.

[72] (0.0764) but in very good agreement with the results of Ref. [73] (0.0710). The right panel shows the residue as a function of ϵ. Good agreement with the analytic theory is obtained provided the finite ϵ effects are kept.

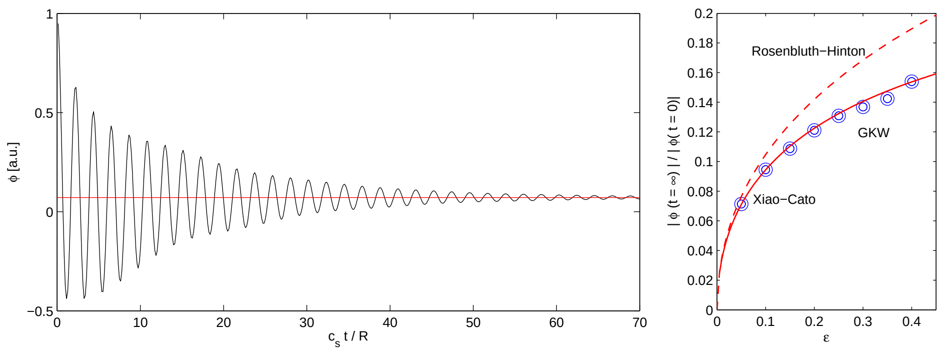


Figure 11.2: Zonal flow benchmark. Left: The potential perturbation (normalised to the potential at $t = 0$) as a function of normalised time, for $q = 1.3$, $\epsilon = 0.05$. The horizontal line gives the residual (0.0711) as predicted in Ref. [73]. For this value of ϵ a numerical residual of 0.0713 is obtained, i.e. in agreement with the analytic result to within 0.3 % Right: Residual $\phi(t = \infty)/\phi(0)$ as a function of ϵ for $q = 1.3$. The circles are the result of GKW, the dotted line is the Rosenbluth-Hinton result [72] and the full line gives the analytic result of Xiao-Catto [73].

The Cyclone base case with adiabatic electrons does not check the correct implementation of the kinetic electron response, and in general is not very sensitive to the effects associated with trapped particles. The physics effects associated with kinetic electrons can be well benchmarked by considering a Trapped Electron Mode. A benchmark with GS2 is shown on the left of Fig. 11.3. The parameters are taken from a discharge in the ASDEX Upgrade tokamak [17]: $\hat{s} = 1.07$, $q = 1.57$, $\epsilon = 0.177$, $T_e/T_i = 3$, $R/L_{Ti} = 0$, electro-static, collisionless, and with an ion to electron mass ratio of a Deuterium plasma. The values of the density gradient as well as the temperature gradient are scanned. The agreement for the growth rates is good.

Finite beta effects are tested for an ITG case increasing the plasma beta. The middle panel of Fig. 11.3 shows the normalised growth rate as a function of the electron beta $\beta = n_e T_e/(B^2/2\mu_0)$ for the Waltz standard case $R/L_{Ti} = R/L_{Te} = 9$, $R/L_n = 3$, $q = 2$ $\hat{s} = 1$, $\epsilon = 0.166$ $T_e/T_i = 1$, and with a mass ratio of a

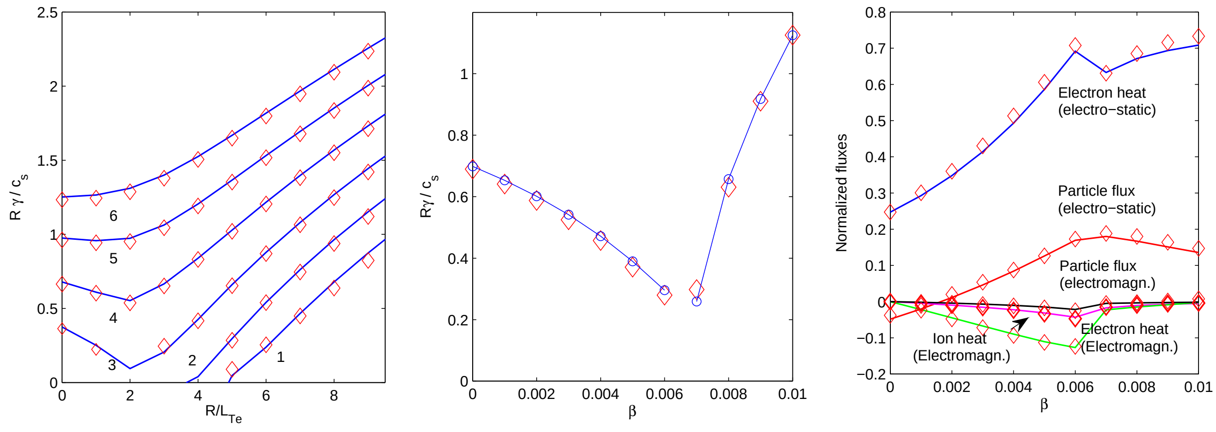


Figure 11.3: Left: Benchmark of a trapped electron mode case. Shown is the normalised growth rate $R\gamma/c_s$ as a function of R/L_{Te} for various values of R/L_n indicated in the figure. Blue lines are the results of GS2 while red diamonds give the results of GKW. Middle: A benchmark of the growth rates of the ITG / Kinetic ballooning mode versus the plasma beta. Blue lines and circles are results from GS2 while red diamonds give the results of GKW. Right: The fluxes, normalised to the ion heat flux as a function of beta. Lines give the results of GS2, while the diamonds give the results of GKW.

Deuterium plasma. The ITG is stabilised by the finite beta effects and, at sufficient high values of beta, a kinetic ballooning mode is destabilised. The agreement for the growth rate between GKW and GS2 is again good. The electro-magnetic case also provides for a good benchmark of the calculation of the fluxes, since all fluxes (also the ones due to the magnetic flutter) are nonzero in this case. The right panel of Fig. 11.3 shows the comparison of the fluxes calculated by GS2 and GKW. Since for a linear problem the amplitude of the mode and, therefore, the magnitude of the fluxes is arbitrary, all fluxes have been normalised to the ion heat flux. This allows for a straightforward comparison between the codes. All calculated fluxes are again in good agreement.

Effects of magnetic field compression are also tested with the linear Waltz standard test case $R/L_{Ti} = R/L_{Te} = 9$, $R/L_n = 3$, $q = 2$, $\hat{s} = 1$, $\epsilon = 0.166$, $T_e/T_i = 1$, the mode with the fastest electrostatic growth rate has been used: $k_\perp \rho_s = 0.3$. Fig. 11.4 shows that the ITG mode is hardly affected by the perturbation $B_{1\parallel}$ but the kinetic ballooning mode is further destabilised with increasing β. Both with the growth rates and the parallel structure a good agreement with GS2 has been obtained.

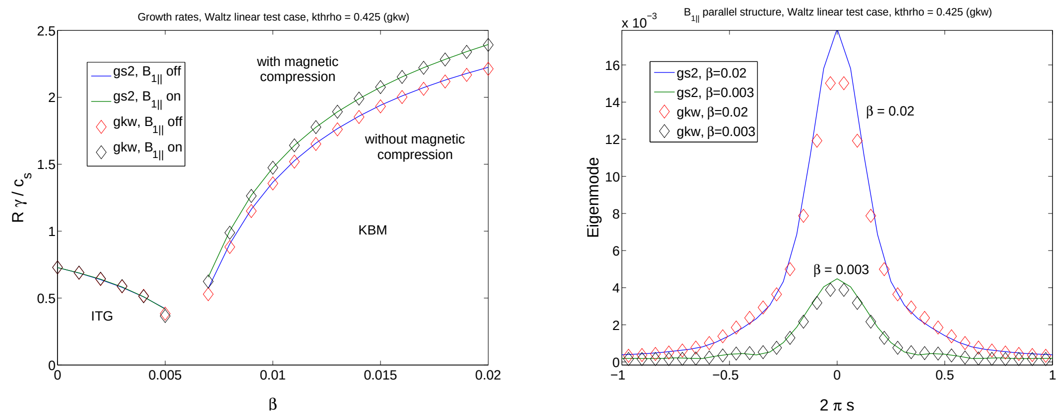


Figure 11.4: Waltz linear test case. Left: Benchmark of growth rates of ITG and kinetic ballooning modes with magnetic field compression on and off. Right: Parallel structure of $B_{1\parallel}$ in ITG and KBM regimes.

A collisionality benchmark of the pitch angle scattering against GS2 for a TEM is presented in Fig. 11.5. To isolate the effect of e-e collisions and e-i collisions, the collisions input variable Z_{eff} (a multiplier for the e-i collision rate) was varied. Scattering of ions (off either species) made no difference for this benchmark (turning them on / off did not change the result of either code). This result was obtained with the Arakawa scheme and zero dissipation in $v_{\|}$. At low collisionalities, GKW convergence for a TEM requires many $v_{\|}$ points (the 64 used for this benchmark are not quite sufficient).

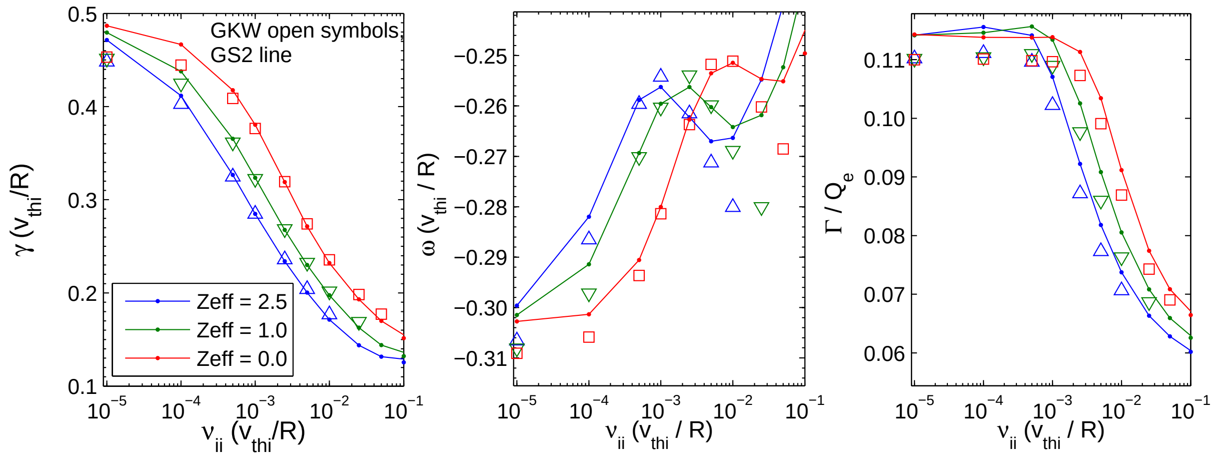


Figure 11.5: Collisions benchmark (pitch angle scattering) for a TEM with varying $Z_{\rm eff}$

The pitch angle collisions have also been benchmarked against GS2 in Ref. [22], with good agreement obtained for the particle flux as a function of collisionality. The treatment of arbitrary toroidal geometry has been benchmarked against GS2 by performing an elongation scan. The parameters considered are $R/L_{Ti} = R/L_{Te} = 8.9$, $R/L_n = 2.85$, $q = 1.42$ $\hat{s} = 1.25$, $\epsilon = 0.182$, $T_e/T_i = 1$ for deuterium ions and adiabatic electrons. The elongation of the last closed flux surface is varied from $\kappa = 1.2$ to $\kappa = 1.6$ and the corresponding MHD equilibrium is calculated using the CHEASE code [42]. This equilibrium is then used for the calculation of the metric tensors both in GKW and GS2. The results for the linear growth rate as a function of $k_\theta \rho_s$ are shown in Fig. 11.6 showing a good agreement between the two codes. The wavevector normalisation (projection) in the two codes in general differs (GKW uses $g^{\zeta\zeta}(LFS)$ while GS2 uses $\mathcal{E}^{\psi\zeta}$ [45] which are only equivalent in '$s-\alpha$' geometry) and must be corrected for (in this case the GKW k_θ inputs were rescaled). Note that the results are significantly different from the ones obtained with the simplified '$s-\alpha$' equilibrium shown by the dashed line (obtained with GKW).

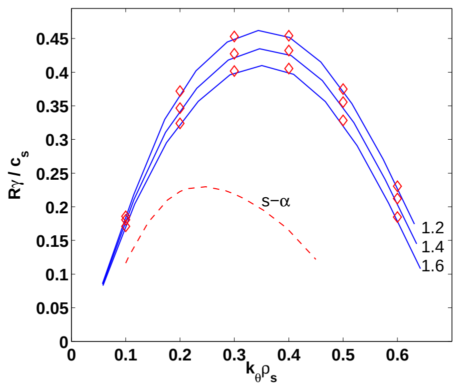


Figure 11.6: Benchmark of the geometry treatment. The normalised growth rate $R\gamma/c_s$ is shown as a function of GS2 $k_\theta\rho_s$ for various values of the last closed flux surface elongation κ indicated in the figure. Blue lines are the results of GS2 while red diamonds give the results of GKW. The results obtained for the simplified '$s-\alpha$' equilibrium with GKW are indicated by the red dashed line.

The effects of impurities have been benchmarked against GS2 using again the Waltz standard test case with '$s-\alpha$' geometry $R/L_{Ti} = R/L_{Te} = 9$, $R/L_n = 3$, $q = 2$, $\hat{s} = 1$, $\epsilon = 0.166$, $T_e/T_i = 1$. Collisions were not included in this test. The left panel of Fig. 11.7 shows the ITG-TEM part of the linear electro-static growth rate spectra for 4 different lithium impurity concentration. The increasing impurity density was balanced by lowering the amount of deuterium in order the preserve quasi-neutrality. This means that the increasing n_{Li} leads to larger values of $Z_{\rm eff}$ which is accountable for the stabilisation of the TEM modes.

On the middle panel of Fig. 11.7 the effect of varying the type of the impurity species can be seen. The impurity density is kept at a constant 1% in this test. Fully ionized lithium, carbon and neon and partially ionized (charge number Z=10 for both cases) iron and tungsten ions have been used. Quasi-neutrality was maintained by adjusting the deuterium density. Here, as well, the increasing Z_{eff} has a stabilizing effect on the TEM modes.

And finally, the right panel of Fig. 11.7 shows the effect of the two electro-magnetic perturbations, the magnetic flutter $A_\parallel$ and the magnetic compression $B_\parallel$, obtained with a lithium impurity density of 10%. The growth rates are significantly increased compared to the electro-static cases. Apart from the longest wavelength modes a good agreement is found between the two codes.

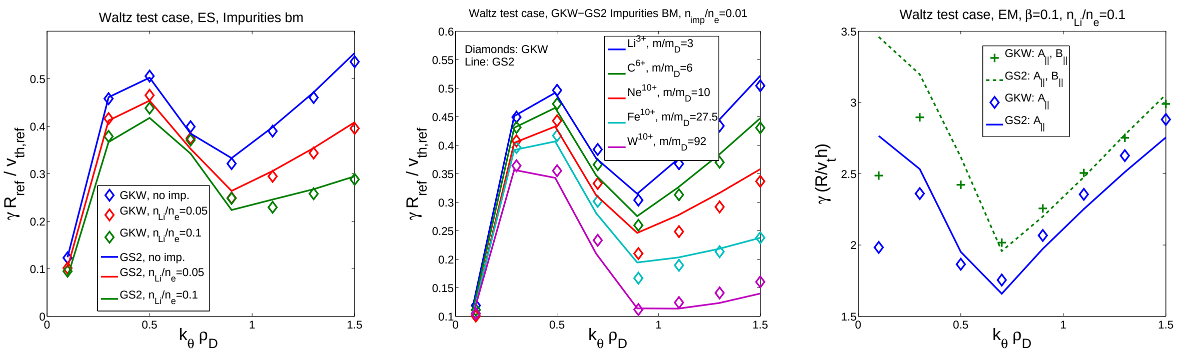


Figure 11.7: Benchmark of impurity treatment, growth rates of the Waltz standard test case with '$s-\alpha$' equilibrium without collisions. Line: GS2, diamonds: GKW. Left panel: effect of varying the Li impurity concentration, electro-static. Middle panel: effect of changing the impurity species, electro-static. Right panel: effect of the electro-magnetic perturations. Green line: fully electro-magnetic, both $A_\parallel$ (magnetic flutter) and $B_\parallel$ (magnetic compression) are retained. Blue line: only $A_\parallel$ is included.

The Miller geometry has been benchmarked against GS2, also for the adiabatic Waltz Standard case, with ϵ changed to 0.266. As for the general geometry benchmark, the different wavevector normalisations must be corrected (in this case the GS2 k_θ inputs were rescaled), after which very good agreement is found (Fig. 11.8.). The squareness and elevation have not been benchmarked since GS2 does not have these parameters. Some of the Miller parameters are defined differently in the two codes and require a simple conversion.

11.2 Nonlinear benchmarks

For the nonlinear solution there are two well established benchmark cases for which several codes have been compared: the cyclone base case, and the Nevins ETG benchmark. GKW has been benchmarked against the European gyrokinetic codes for the Cyclone case in Ref. [49].

Here we describe the alternative ETG benchmark of Nevins [75]. The parameters for this case are the same as those of the Cyclone base case ($R/L_T = 6.9$, $R/L_n = 2.2$, $q = 1.4$, $T_e/T_i = 1$, $\epsilon = 0.18$) with only the magnetic shear changed to $\hat{s} = 0.1$. For this case the electron dynamics is simulated while the ions are assumed adiabatic. The latter response does not include the flux surface average of the electro-static potential, i.e. the response is proportional to $n_i\phi/T_i$. To obtain the same results as published in Ref. [75] we have used (and must use) the same number of bi-normal modes $N_{\rm mod} = 8$, and mode spacing $(k_\perp\rho_s)_{\max} = 0.69$. The radial grid spacing is $\Delta(k_r\rho_s) = 0.0619$ with $N_x = 41$ (both positive and negative

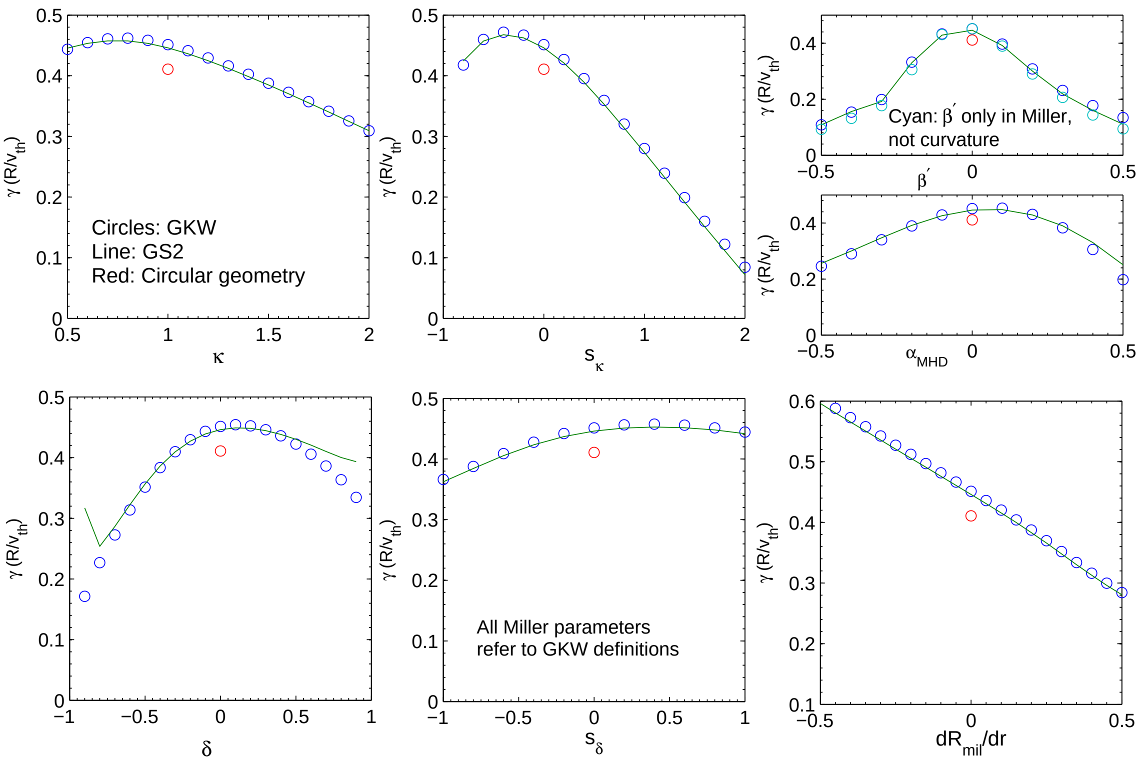


Figure 11.8: Benchmark of the Miller geometry for the growth rates of the adiabatic Waltz standard case with $\epsilon = 0.266$ and GKW $k_\theta \rho_i = 0.43$. Line: GS2, Circles: GKW. The Miller parameters shown on the horizontal axes refer to the GKW definitions.

wave vectors counted). Finally, $N_\mu = 8$, $N_{v_\parallel} = 16$, and $N_s = 16$.

Figure 11.9 shows the normalised electron heat flux as a function of the normalised time. Averaging over the time interval ($1000 < tc_e/L_T < 19500$), GKW predicts a heat diffusion coefficient $\chi_e = 3.08 \pm 0.19$ in good agreement with the value $\chi_e = 2.95 \pm 0.15$ reported in Ref. [75]. The ETG case is sensitive to the magnetic shear. For values $\hat{s} > 0.3$ no convergence is found in the simulations reported in Ref. [75]. GKW reproduces this result as shown in Fig. 11.10.

The background ExB shear stabilisation has been successfully benchmarked against GYRO results as reported in Ref. [24]. This result for the Waltz standard case is included again here for convenience in Fig. 11.11. GKW has also been checked against the GS2 results reported in Ref. [50] for the Cyclone case, also shown in Fig. 11.11. Both of these comparisions were against existing results, i.e. no further attempt to examine the convergence of the GYRO and GS2 results has been made. Detailed convergence testing for the GKW results is described in Refs. [51] and [52]. This study revealed that it is preferable to use some radial hyperdissipation and minimal parallel dissipation at high shear rates (see also Sec. 9.3.2.) (not the case for these figures) which can improve the agreement at high shear rates to better than that shown in Fig. 11.11.

Towards the edge, nonlinear and electromagnetic physics can be significantly more complex than the ITG dominated Cyclone and GA-STD cases, and benchmarking in this area is an active area of research. For L-mode ASDEX-Upgrade plasmas, GKW has been sucessfully benchmarked with GENE for four nonlinear electromagnetic cases using full geometry, and full collisions. The results are reported in in Refs. [54] and [55]. Benchmarking for the the DIII-D ‘shortfall’ case is in progress and is reported in ITPA presentations of C. Holland and Y. Camenen.

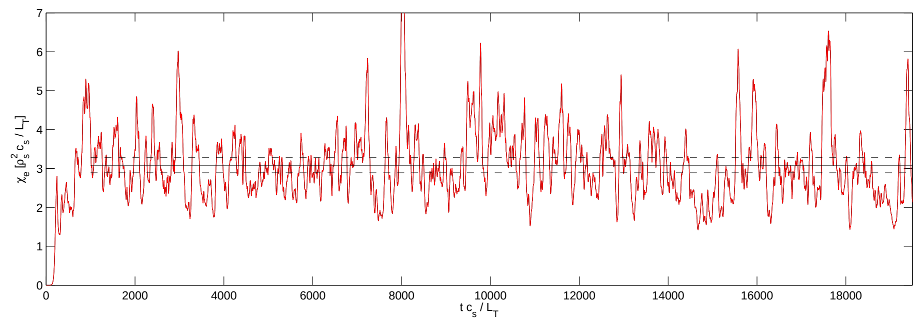


Figure 11.9: GKW results of the electron heat flux expressed as a heat conduction coefficient (normalised to $\rho_e^2 c_e/L_T$ with $c_e = \sqrt{T_e/m_e}$) as a function of normalised time ($t_{NN} = tc_e/L_T$) for the Nevins ETG benchmark

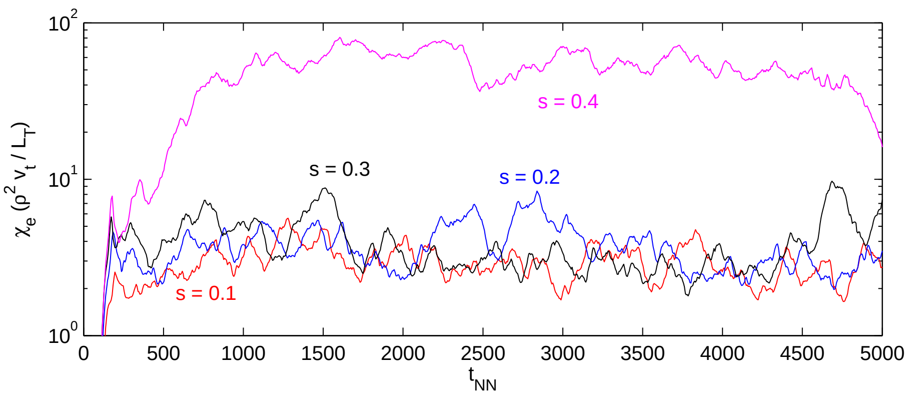
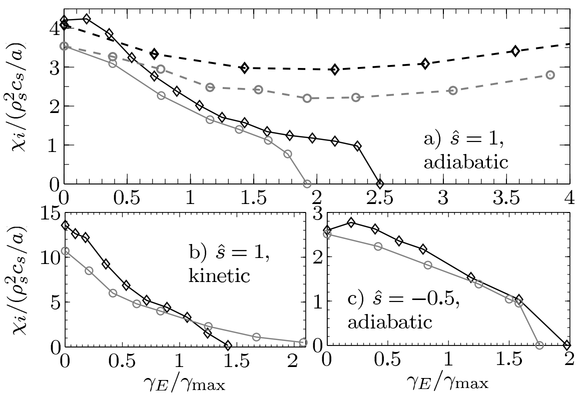


Figure 11.10: GKW results of the electron heat conduction coefficient for different values of the magnetic shear

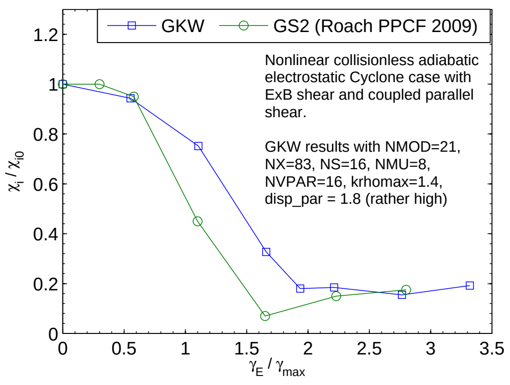
11.2.1 Global Benchmarks

Can be found in the Ringberg 2012 presentation of A.G Peeters on the GKW webpages under Talks..

Acknowledgements The authors want to thank W. Dorland and M. Kotschenreuther for making the GS2 code available, as well as A. Marinoni for his contribution to the geometry benchmark. One of the authors (AGP) would also like to thank Bruce D. Scott for his tutorials and insights into the subject which he has been able to follow over many years.



(a) Benchmark with GYRO



(b) Benchmark with GS2

Figure 11.11: Left: GKW benchmark of background ExB shearing with GYRO for Waltz standard case parameters. GKW results (diamonds) for adiabatic electrons, kinetic electrons, and adiabatic with $\hat{s} = -0.5$, are compared to the equivalent GYRO results (circles) from Tables II, III, IV and V of Ref. [53]. The dashed lines include coupled parallel velocity shear for purely toroidal rotation with $u' = 12\gamma_E$. Right: GKW benchmark of background ExB shearing with GS2 results reported in Ref. [50] for Cyclone case parameters. Both codes were run including coupled parallel velocity shear for purely toroidal rotation. $\gamma_{\max}$ is the maximum linear growth rate at $\gamma_E = 0$.

Appendix A

Geometry - generalised

A.1 Field aligned Hamada coordinates

In this Appendix the straight field line coordinates as well as the shifted metric are described in detail. The development of the material in this section has benefited from, and closely follows the publications of B.D. Scott [40, 41], and the references cited therein. This material is in addition to Section 2.2 of this document (which describes the specific geometry models implemented); here we add a generalised formulation for deriving the geometry tensors, and describe the shifted metric procedure for the nonspectral version of the code. At some point in the future the two sections should be consolidated.

We start with an arbitrarily shaped, but toroidally symmetric geometry, for which the magnetic field can be written in the form

$$\mathbf{B} = s_B F \nabla\varphi + s_j \nabla\varphi \times \nabla\Psi, \tag{A.1}$$

where $F = RB_t$ with $B_t > 0$ being the toroidal magnetic field strength. The quantities s_B and s_j represent the sign of the magnetic field and plasma current (both positive when in the direction of $\nabla\varphi$. Finally, Ψ is the poloidal flux, not to be confused with the radial coordinate (ψ). Of course, $\psi = \psi(\Psi)$, and the choice $\psi = \Psi$ will sometimes be made below.

An orthogonal coordinate system (ψ, θ, φ) is assumed, where ψ is the 'radial' coordinate (i.e. $\mathbf{B} \cdot \nabla\psi = 0$), θ is the poloidal angle (upward on the outboard midplane), and φ is the toroidal angle (clockwise when viewed from above). Furthermore, it is assumed that the magnetic field and quantities like the major radius are known in this coordinate system. Below two coordinate transformations are introduced. The first will make the field lines straight in the new coordinates, and the second will align one of the coordinates with the magnetic field.

In the first step we transform the poloidal and toroidal angle

$$s = s(\psi, \theta), \qquad \gamma = \gamma(\psi, \theta, \varphi) \tag{A.2}$$

in such a way that the contra-variant components of the magnetic field are flux functions in the new coordinate system ($B^s = B^s(\psi)$, $B^\gamma = B^\gamma(\psi)$). These coordinates are known as Hamada coordinates [36]. Note that the new coordinate γ is still an ignorable coordinate, since any function that is independent of φ will be independent of γ ($f(\theta, \psi) \to f(s, \psi)$) after the coordinate transformation. The transformation of the poloidal angle is chosen such that B^s becomes a flux function

$$B^s = \mathbf{B} \cdot \nabla s = \mathbf{B} \cdot \nabla\theta \frac{\partial s}{\partial \theta} + \mathbf{B} \cdot \nabla\psi \frac{\partial s}{\partial \psi} = \mathbf{B} \cdot \nabla\theta \frac{\partial s}{\partial \theta} \tag{A.3}$$

From which the relation between s and θ can be derived

$$\frac{\partial s}{\partial \theta} = \frac{B^s}{\mathbf{B} \cdot \nabla\theta} \Leftrightarrow \int_0^\theta \frac{\partial s}{\partial \theta} d\theta = s = B^s \int_0^\theta \frac{d\theta}{\mathbf{B} \cdot \nabla\theta} \tag{A.4}$$

since B^s does not depend on θ (flux function) and

$$\oint \frac{\partial s}{\partial \theta} d\theta = 1 = B^s \oint \frac{d\theta}{\mathbf{B} \cdot \nabla \theta} \Leftrightarrow B^s = \frac{1}{\oint \frac{d\theta}{\mathbf{B} \cdot \nabla \theta}} \tag{A.5}$$

which gives the following equation for s:

$$s(\theta, \psi) = \int_0^\theta \frac{\mathrm{d}\theta'}{\mathbf{B} \cdot \nabla \theta'} \bigg/ \oint \frac{\mathrm{d}\theta'}{\mathbf{B} \cdot \nabla \theta'} \qquad \text{with} \qquad B^s = 1 \bigg/ \oint \frac{\mathrm{d}\theta'}{\mathbf{B} \cdot \nabla \theta'} \tag{A.6}$$

where the normalizing constant has been chosen such that the domain [-1/2,1/2] in s corresponds to one poloidal turn.

The new coordinate is closely related to the flux surface average. This average is defined as the volume average between two flux surfaces in the limiting case in which the distance between the flux functions goes to zero. One can derive the following relation

$$\{g\} = \oint \frac{\mathrm{d}l_p}{B_p} g \bigg/ \oint \frac{\mathrm{d}l_p}{B_p} \tag{A.7}$$

where l_p is the length in the poloidal direction ($\mathrm{d}l_p = |\mathbf{e}_\theta| \mathrm{d}\theta$), B_p is the poloidal component of the magnetic field strength ($B_p = \mathbf{B} \cdot \nabla \theta / |\nabla \theta|$), and the integral is to be taken at constant ψ. Inserting the expressions for the poloidal length and the poloidal magnetic field strength and using $|\mathbf{e}_\theta||\nabla \theta| = 1$ one obtains

$$\{g\} = \oint \mathrm{d}s g \tag{A.8}$$

For the transformation of the toroidal angle we choose

$$\gamma = \frac{\varphi}{2\pi} + g(\theta, \psi) \tag{A.9}$$

The 2π is added here such the domain [0,1] in γ will correspond to one toroidal turn around the flux surface. The contra-variant component of the magnetic field is

$$B^\gamma = s_B \frac{F}{2\pi} \frac{1}{R^2} + \mathbf{B} \cdot \nabla \theta \frac{\partial g}{\partial \theta} \tag{A.10}$$

where $F = RB_t = F(\psi)$ is a flux function. (Here B_t is the toroidal magnetic field strength $B_t = \mathbf{B} \cdot \nabla \varphi / |\nabla \varphi|$ where $|\nabla \varphi| = 1/R$ with R being the local major radius) From the expression above one can derive

$$g(\theta, \psi) = \int_0^\theta \frac{\mathrm{d}\theta'}{\mathbf{B} \cdot \nabla \theta'} \left[B^\gamma - s_B \frac{F}{2\pi} \frac{1}{R^2} \right] \tag{A.11}$$

Demanding that g is periodic

$$\oint \frac{\mathrm{d}\theta'}{\mathbf{B} \cdot \nabla \theta'} \left[B^\gamma - s_B \frac{F}{2\pi} \frac{1}{R^2} \right] = 0 \tag{A.12}$$

yields

$$B^\gamma = s_B \frac{F}{2\pi} \left\{ \frac{1}{R^2} \right\} \tag{A.13}$$

The coordinate γ can then be expressed as

$$\gamma = \frac{\varphi}{2\pi} + s_B \frac{F}{2\pi} \int_0^\theta \frac{\mathrm{d}\theta'}{\mathbf{B} \cdot \nabla \theta'} \left[\left\{ \frac{1}{R^2} \right\} - \frac{1}{R^2} \right] \tag{A.14}$$

The coordinates (ψ, s, γ) are the Hamada coordinates. The expression above should allow these coordinates to be calculated for an arbitrary toroidally symmetric geometry.

We now go through the second coordinate transformation where one of the coordinates is aligned with the magnetic field, i.e. we demand

$$\mathbf{B} \cdot \nabla = B^s \frac{\partial}{\partial s} \tag{A.15}$$

Note that this does not mean that ∇s is in the direction of the magnetic field. In general it is not. The transformation is

$$\zeta = \zeta(s, \gamma, \psi) \tag{A.16}$$

In the new coordinates

$$\mathbf{B} \cdot \nabla = B^\psi \frac{\partial}{\partial \psi} + B^s \frac{\partial}{\partial s} + B^\zeta \frac{\partial}{\partial \zeta} \tag{A.17}$$

where B^ψ is zero. The transformation to field aligned coordinates is the transformation for which $B^\zeta = 0$. The latter condition can be formulated as

$$B^\gamma \frac{\partial \zeta}{\partial \gamma} + B^s \frac{\partial \zeta}{\partial s} = 0 \tag{A.18}$$

which is satisfied for the simple linear transformation

$$\zeta = qs - \gamma \qquad \text{where} \qquad q = \frac{B^\gamma}{B^s} \tag{A.19}$$

From the field line equation it follows that

$$\frac{\mathrm{d}\gamma}{\mathrm{d}s} = \frac{B^\gamma}{B^s} \qquad \rightarrow \qquad \oint \frac{\mathrm{d}\gamma}{\mathrm{d}s} \mathrm{d}s = \oint \frac{B^\gamma}{B^s} \mathrm{d}s = q \tag{A.20}$$

i.e. $q = q(\psi)$ is indeed the safety factor.

Note that the coordinate transformation above flips the sign of the toroidal angle. The right handed coordinate system can therefore be defined as (ψ, ζ, s). The Jacobian of the new coordinate system can be expressed in terms of the original Jacobian through

$$(\nabla \psi \times \nabla \zeta) \cdot \nabla s = \frac{1}{J_{\psi \zeta s}} = \frac{1}{2\pi} \frac{\partial s}{\partial \theta} \frac{1}{J_{\psi \theta \varphi}} \tag{A.21}$$

From which it follows

$$J_{\psi \zeta s} = 2\pi \mathbf{B} \cdot \nabla \theta \oint \frac{\mathrm{d}\theta'}{\mathbf{B} \cdot \nabla \theta'} J_{\psi \theta \varphi} = 2\pi \frac{\mathbf{B} \cdot \nabla \theta}{B^s} J_{\psi \theta \varphi} \tag{A.22}$$

We note here that the coordinates s and ζ are dimensionless, but that the dimension of ψ is not yet defined. We will always use a normalized ψ that is dimensionless.

A.2 Caculating the Geometry tensors from the metric tensor and derivatives of coordinates and magnetic field

The various tensors that are used for the implementation of the geometry are not independent. $\mathcal{E}$, $\mathcal{D}$, $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, $\mathcal{I}$, $\mathcal{J}$, $\mathcal{K}$ can be expressed in the metric tensor and derivatives of the coordinates as well as the magnetic field. We will assume here that the metric tensor $g^{\alpha\beta}$, the magnetic field strength (B) and its derivatives towards the coordinates ($\partial B/\partial s, \partial B/\partial \psi$), the major radius R and its derivatives ($\partial R/\partial s, \partial R/\partial \psi$), the z-coordinate Z and its derivatives ($\partial Z/\partial s, \partial Z/\partial \psi$) and finally the derivative of the poloidal flux towards the radial coordinate $\partial \Psi/\partial \psi$ are given. Of course, even these quatities are not independent, since the derivatives of R, Z towards the coordinates can be used to calculate the metric tensor. The tensors can then be calculated as discussed below

Since we can write any vector $\mathbf{A}$ as

$$\mathbf{A} = \frac{1}{2}\epsilon_{ijk} A^i J_{x^i x^j x^k} \nabla x^j \times \nabla x^k, \tag{A.23}$$

one finds for the magnetic field, using $\psi = \Psi$,

$$\mathbf{B} = B^s J_{\Psi\zeta s} \nabla\Psi \times \nabla\zeta. \tag{A.24}$$

Using

$$B^s = \mathbf{B}\cdot\nabla s = s_b F \nabla\varphi \cdot \nabla s + s_j \nabla\varphi \times \nabla\Psi \cdot \nabla s \tag{A.25}$$

and remembering that by definition of ζ

$$\nabla\varphi \times \nabla\Psi \cdot \nabla s = -2\pi \nabla\zeta \times \nabla\Psi \cdot \nabla s, \tag{A.26}$$

one gets

$$B^s = 2\pi s_j \frac{1}{J_{\Psi\zeta s}} \tag{A.27}$$

Finally, one obtains

$$\mathbf{B} = 2\pi s_j \nabla\Psi \times \nabla\zeta \tag{A.28}$$

The Clebsch representation of the magnetic field can directly be used to calculate the tensor $\mathcal{E}$

$$\mathcal{E}^{\alpha\beta} = \frac{1}{2B^2}\mathbf{B}\cdot(\nabla x^\alpha \times \nabla x^\beta) = \frac{\pi s_j}{B^2}\frac{\partial\Psi}{\partial\psi}[g^{\psi\alpha}g^{\zeta\beta} - g^{\zeta\alpha}g^{\psi\beta}] \tag{A.29}$$

With the $\mathcal{E}$ tensor and the derivatives of the magnetic field strength given, the $\mathcal{D}$ vector directly follows from

$$\mathcal{D}^\alpha = -2\mathcal{E}^{\alpha\beta}\frac{1}{B}\frac{\partial B}{\partial x_\beta} \tag{A.30}$$

The scalars that have been given earlier in this document are

$$\mathcal{F} = \frac{B^s}{B} \qquad \mathcal{G} = \frac{\mathcal{F}}{B}\frac{\partial B}{\partial s} \tag{A.31}$$

The angular rotation vector points in the ∇Z direction, and

$$\mathbf{\Omega}_\perp = \Omega \mathbf{b} \times (\nabla z \times \mathbf{b}) \tag{A.32}$$

This allows the $\mathcal{H}$ vector to be written in the form

$$\mathcal{H}^\alpha = -\frac{s_B}{B_N}\left[\frac{\partial Z}{\partial x_\beta}g^{\alpha\beta} - \left(\frac{B^s}{B}\right)^2 \frac{\partial Z}{\partial s}\delta_{\alpha s}\right] \tag{A.33}$$

The $\mathcal{I}$ tensor is straightforwardly

$$\mathcal{I}^\alpha = \frac{1}{2B}(\nabla x_\alpha \times \nabla R^2)\cdot\mathbf{b} = 2R\mathcal{E}^{\alpha\beta}\frac{\partial R}{\partial x_\beta} \tag{A.34}$$

The $\mathcal{J}$ and $\mathcal{K}$ tensor can be directly calculated from the known quantities.

A.3 Periodicity

The torus is periodic in both the toroidal as well as the poloidal angle. In the new coordinates this means that for any function $f(\psi, \zeta, s)$

$$f(\psi, \zeta + 1, s) = f(\psi, \zeta, s) \tag{A.35}$$

$$f(\psi,\zeta+q,s+1) = f(\psi,\zeta,s) \tag{A.36}$$

In a flux tube one, furthermore, demands that the solution is periodic in the radial direction

$$f(\psi+L_\psi,\zeta,s) = f(\psi,\zeta,s) \tag{A.37}$$

None of the equilibrium quantities is a function of ζ. For these quantities Eq. (A.35) is trivially satisfied, and the Eq. (A.36) reduces to

$$B(\psi,s+1) = B(\psi,s) \tag{A.38}$$

where we have use the background magnetic field strength as example. The perturbed quantities (distribution function as well as fields) for the $\zeta-$ direction are always represented by Fourier modes

$$f(\psi,\zeta,s) = \sum\nolimits_{k_\zeta} \hat{f}(\psi,k_\zeta,s)\exp[\mathrm{i}k_\zeta\zeta/\rho_*] \tag{A.39}$$

The periodicity constraint of Eq. (A.36) then reduces to

$$\hat{f}(\psi,k_\zeta,s+1) = \hat{f}(\psi,k_\zeta,s)\exp[-\mathrm{i}k_\zeta q/\rho_*] \tag{A.40}$$

The flux tube treats the limit of $\rho_* \to 0$ to the lowest relevant order only. Even though the phase factor can in principle take any value, we can assume that a very small change in the safety factor can always be choosen such that

$$\exp[-\mathrm{i}k_\zeta q/\rho_*] = 1 \tag{A.41}$$

without changing the results. This choice can, however, only be made for one location since the safety factor is a function of the radial coordinate ψ. We choose the phase factor above to be zero for the centre of the box (presented by $\psi = 0$ to obtain a periodicity constraint

$$\hat{f}(\psi k_\zeta,s+1) = \hat{f}(\psi,k_\zeta,s)\exp\left[-\mathrm{i}k_\zeta\frac{1}{\rho_*}\frac{\mathrm{d}q}{\mathrm{d}\psi}\psi\right] \tag{A.42}$$

A.4 Shifted metric

In general the metric tensor has no zero elements. Especially the magnetic shear leads to a finite $g^{\zeta\psi}$ element that is a function of the coordinate along the field line. This means that an originally rectangular grid cell, deforms when moving along the magnetic field. This can potentially lead to problems [41]. A remedy, that does not cost anything computationally when using Fourier modes in ζ is the shifted metric procedure. In this procedure we change the coordinate ζ at every location in s.

$$\zeta_k = qs - \gamma - \alpha_k(\psi) \tag{A.43}$$

It is important that α is not a function of ζ. With this transformation one has a different set of coordinates at every grid point in s, with the different grid points denoted by the index k. The metric tensor in the new coordinates can be expressed as

$$g_k^{\zeta\zeta} = g^{\zeta\zeta} - 2\alpha'_k g^{\zeta\psi} + (\alpha'_k)^2 g^{\psi\psi} \tag{A.44}$$

$$g_k^{\zeta\psi} = g^{\zeta\psi} - \alpha'_k g^{\psi\psi} \tag{A.45}$$

$$g_k^{\zeta s} = g^{\zeta s} - \alpha'_k g^{\psi s} \tag{A.46}$$

where $\alpha'_k = \partial\alpha_k/\partial\psi$. With this transformation one can set the $g^{\zeta\psi}$ element at the grid point $s = s_k$ locally to zero

$$\alpha'_k = \frac{g^{\zeta\psi}(s_k)}{g^{\psi\psi}(s_k)} \tag{A.47}$$

In the flux tube the metric elements are evaluated at one surface only, and are not a function of the radial coordinate. The equation above can then be trivially integrated to obtain

$$\alpha_k = \frac{g^{\zeta\psi}(s_k)}{g^{\psi\psi}(s_k)}\psi \tag{A.48}$$

In the new coordinates

$$g_k^{\zeta\zeta} = g^{\zeta\zeta} - (g^{\zeta\psi})^2/g^{\psi\psi} \tag{A.49}$$

$$g_k^{\zeta\psi} = 0 \tag{A.50}$$

$$g_k^{\zeta s} = g^{\zeta s} - g^{\zeta\psi}g^{\psi s}/g^{\psi\psi} \tag{A.51}$$

The perturbed solution in the new coordinates has the form

$$\hat{f}_k(\psi, k_\zeta, s)\exp[\mathrm{i}k_\zeta\zeta_k/\rho_*] \tag{A.52}$$

We must demand that the solution is single valued, and therefore

$$\hat{f}_k(\psi, k_\zeta, s)\exp[\mathrm{i}k_\zeta\zeta_k/\rho_*] = \hat{f}_p(\psi, k_\zeta, s)\exp[\mathrm{i}k_\zeta\zeta_p/\rho_*] \tag{A.53}$$

Using

$$\zeta_p = \zeta_k + \alpha_k - \alpha_p \tag{A.54}$$

We obtain

$$\hat{f}_k(\psi, k_\zeta, s) = \hat{f}_p(\psi, k_\zeta, s)\exp[\mathrm{i}k_\zeta(\alpha_k - \alpha_p)/\rho_*] \tag{A.55}$$

The perturbed distribution function as well as the fields are calculated at the discrete points s_k as f_k. When taking a parallel derivative one must use the transformation given above to transform $\hat{f}_p$ to $\hat{f}_k$.

Finally we must consider the possible radial periodicity of the domain. In the new coordinates Eq. (A.37) becomes

$$f_k(\psi + L_\psi, \zeta_k + \alpha_k'\psi + \alpha_k'L_\psi, s) = f_k(\psi, \zeta_k + \alpha_k'\psi, s) \tag{A.56}$$

Or

$$\hat{f}_k(\psi + L_\psi, k_\zeta, s) = \hat{f}_k(\psi, k_\zeta, s)\exp[-\mathrm{i}k_\zeta\alpha_k'L_x] \tag{A.57}$$

A.5 Summary of the sign conventions in GKW

In GKW, the cylindrical coordinate system (R, Z, φ) is right handed, which means that φ is increasing clockwise when the torus is viewed from above.
The toroidal rotation is defined positive for a plasma flowing in the direction of **B**.
The mode frequency is defined positive for a perturbation evolving in the direction opposite to $\nabla\zeta$. This corresponds to the ion ∇B drift direction if $s_{\mathrm{j}} = 1$ and to the electron ∇B drift direction if $s_{\mathrm{j}} = -1$. See also Fig. 10.1
Coordinate system:

- $\psi = \varepsilon = (R_{\max} - R_{\min})/(2R_{\mathrm{ref}})$ is always increasing from the plasma center to the plasma edge.
- s is always increasing upwards from the low field side midplane. It is zero at the height of the magnetic axis, on the low field side midplane.

- ζ is always increasing in the direction opposite to φ (i.e. anticlockwise when viewed from above) at constant ψ and s. The direction of $\nabla\zeta$ in the poloidal plane is given by $\text{sign}(\nabla s\cdot\nabla\zeta) = \text{sign}(\nabla\zeta\cdot\nabla\theta) = s_\text{B}s_\text{j}$

$$\begin{aligned}
\text{sign}(\mathbf{B}\cdot\nabla\varphi) &\equiv s_B & \text{(A.58)}\\
\text{sign}(\mathbf{j}\cdot\nabla\varphi) &\equiv s_j & \text{(A.59)}\\
\text{sign}(\mathbf{B}\cdot\nabla\theta) &= s_j & \text{(A.60)}\\
\text{sign}(\mathbf{B}\cdot\nabla s) &= s_j & \text{(A.61)}\\
\text{sign}(\nabla\varphi\cdot\nabla\zeta) &= -1 & \text{(A.62)}\\
\text{sign}(\nabla s\cdot\nabla\theta) &= 1 & \text{(A.63)}\\
\text{sign}(\nabla\theta\cdot\nabla\zeta) &= s_B s_j & \text{(A.64)}\\
\text{sign}(B_\theta\nabla\theta\cdot\nabla\zeta) &= s_B & \text{(A.65)}\\
\text{sign}(\nabla s\cdot\nabla\zeta) &= s_B s_j & \text{(A.66)}\\
\mathbf{B}\cdot\nabla\zeta &= 0 & \text{(A.67)}\\
\mathbf{\Omega} &= -s_B\Omega\nabla z & \text{(A.68)}\\
u &= R\Omega & \text{(A.69)}
\end{aligned}$$

A.6 Flux surface average

A.6.1 General

The general definition of the flux surface average is

$$< A >= \frac{\int A(\mathbf{x})\delta(r-r_0)d^3x}{\int \delta(r-r_0)d^3x} \qquad \text{(A.70)}$$

where A is the quantity being averaged, r_0 is the flux surface label of flux surface on which the average is being performed, r is the value of the flux surface label at position $\mathbf{x}$, δ is the Dirac function and the integral is being performed over the whole plasma (or entire world, it does not matter) volume.

A.6.2 GKW coordinates

In GKW coordinates (r, ζ, s), the elementary plasma volume is

$$d^3x = drd\zeta ds\mathcal{J} \qquad \text{(A.71)}$$

with $\mathcal{J}^{-1} = \nabla r\cdot\nabla\zeta\times\nabla s$. Actually the expression above is true for any coordinates, but one of the specificity of GKW coordinate system is that $\mathcal{J}$ is a flux surface label:

$$\mathcal{J} = \frac{\partial\Psi}{\partial r}\frac{s_j}{B^s} \qquad \text{(A.72)}$$

and B^s is a flux surface label by construction. This implies that the flux surface average can be written as

$$< A >= \frac{\oint_{r=r_0} A(r,\zeta,s)d\zeta ds}{\oint_{r=r_0} d\zeta ds} \qquad \text{(A.73)}$$

If A is independent of the toroidal angle, one arrives at

$$< A >= \oint A(r_0, s)ds \qquad \text{(A.74)}$$

A.6.3 Alternative expression

Noting that the surface element $\mathrm{d}S$ on a flux surface is related to the volume element d^3x by

$$\mathrm{d}^3x = \mathrm{d}S\frac{\mathrm{d}r}{|\nabla r|} \tag{A.75}$$

with r an arbitrary flux surface label, an alternative expression for the flux surface average is obtained:

$$< A >= \frac{1}{V'}\oint A\frac{\mathrm{d}S}{|\nabla r|} \tag{A.76}$$

with

$$V' = \frac{\partial V}{\partial r} = \oint \mathcal{J}\mathrm{d}\zeta\mathrm{d}s \tag{A.77}$$

and V being the volume enclosed by the flux surface labelled by r.

A.7 Use of fluxes in transport equations

In the following the density conservation equation is taken as an example, but the same can be applied to heat, momentum or any conserved quantity. The local density conservation equation is

$$\frac{\partial n}{\partial t} + \nabla\cdot\mathbf{\Gamma} = S_n \tag{A.78}$$

with n the density in m^{-3}, $\mathbf{\Gamma}$ the particle flux in $\mathrm{m}^{-2}\mathrm{s}^{-1}$ and S_n the particle source density in $\mathrm{m}^{-3}\mathrm{s}^{-1}$

Integrating this equation over the volume enclosed by a flux surface labelled by r_0 leads to:

$$\frac{\partial}{\partial t}\int n\mathrm{d}V + \oint\mathbf{\Gamma}\cdot\frac{\nabla r}{|\nabla r|}\mathrm{d}S = \int S_n\mathrm{d}V \tag{A.79}$$

where the flux surface shape and position has been assumed to be constant in time. Using Eq. (A.76), the continuity equation can then be written as

$$\frac{\partial}{\partial t}\int n\,\mathrm{d}V + V'\left\langle\mathbf{\Gamma}\cdot\nabla\mathbf{r}\right\rangle = \int S_n\mathrm{d}V \tag{A.80}$$

In steady-state we have:

$$< \Gamma^r >=< \mathbf{\Gamma}\cdot\nabla\mathbf{r} >= \frac{1}{V'}\int S_n\mathrm{d}V = \frac{1}{V'}S_n^{\mathrm{int}} \tag{A.81}$$

This means that Γ^r which is the quantity we calculate in GKW needs to be compared to the particle source inside the flux surface S_n^{int} divided by V'. This is the obvious choice for comparisons with the experiments or transport codes.

Concerning the flux decomposition, the best choice is a direct decomposition of $< \Gamma^r >$:

$$< \Gamma^r >= -D\frac{\partial n}{\partial r} + Vn \tag{A.82}$$

with possibly a normalising length if r is a dimensionless flux label (e.g. GKW ψ). For a benchmark or a comparison with experiment one needs to make sure that everybody uses the same definition.

Appendix B

Transformation of transport coefficients in presence of poloidal asymmetries

Memo on the transformation of the transport coefficients from LFS density to flux-surface averaged density

C. Angioni, E. Belli and F.J. Casson

Please report comments and corrections to C. Angioni

Note: The notation used in this Appendix does not attempt to be consistent with the rest of the GKW manual.

In the presence of a poloidally asymmetric density $n(r,\theta)$ of a particle species like W, the flux surface averaged particle flux Γ can be expressed as a function of the flux surface averaged density $n(r) = \langle n(r,\theta)\rangle$ or as a function of densities evaluated at specific locations or poloidal angles. A natural choice, for instance, is $n_0(r) = n(r,\theta = 0)$, evaluated at the LFS $\theta = 0$ location, which is adopted by codes like GKW and NEO.

In the present memo we provide the equations to transform the corresponding transport coefficients from one description to the other. This is practical for many applications, in particular when the output of codes which can include 2D poloidal asymmetries has to be used inside usual 1D transport codes, which consider only 1D (or in any case flux surface averaged) quantities.

The flux surface averaged particle flux is expressed in the two forms

$$\Gamma = -D\frac{dn}{dr} + Vn,$$

and

$$\Gamma = -D_0\frac{dn_0}{dr} + V_0 n_0.$$

and we state that these two forms are equivalent. In the present memo, we shall made this equivalence explicit.

We note that the first expression for Γ involves the flux surface averaged density $n(r) = \langle n(r,\theta)\rangle$, and the corresponding transport coefficients D and V. In contrast, the second expression involves the LFS density $n_0 = n(r,\theta = 0)$ and the corresponding transport coefficients D_0 and V_0.

We remind that in general

$$n(r,\theta) = n_0(r)\,\exp\left\{-\frac{Ze\Phi(r,\theta)}{T(r)} + \frac{m\Omega^2(r)}{2T(r)}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\}$$

where Φ is the background electrostatic potential, defined in such a way that $\Phi(r, \theta = 0) = 0$, and $R_0(r)$ is the major radius of the location where n_0 is evaluated (usually LFS $\theta = 0$).

We introduce also the auxiliary (species dependent) normalized energy

$$E(r,\theta) = \frac{Ze\Phi(r,\theta)}{T(r)} - \frac{m\Omega^2(r)}{2T(r)}\left(R(r,\theta)^2 - R_0(r)^2\right)$$

For clarity, with circular concentric flux surfaces, $R(r,\theta) = R_{\text{geo}} + r\cos\theta$ and $R_0(r) = R_{\text{geo}} + r$.

In general flux tube geometry, θ is a generalized poloidal angle, which describes the distance along the field line. Considering the right handed system of coordinates (r, θ, ϕ), the flux surface average of the quantity $n(r, \theta)$ is given by

$$\langle f\rangle = \frac{1}{V'}\oint d\theta d\phi\, \sqrt{g}\, n(r,\theta),$$

where $g = (\nabla r \times \nabla\theta \cdot \nabla\phi)^{-2}$ is the determinant of the metric tensor, and $V' = dV/dr$, where $V(r)$ is the plasma volume up to the flux surface r.

Then, the following relationship holds,

$$\frac{d\langle n(r,\theta)\rangle}{dr} = \left\langle \frac{\partial n(r,\theta)}{\partial r}\right\rangle + n_0(r)\,\mathcal{G}_r,$$

where we have defined

$$\mathcal{G}_r = \left[-\frac{1}{V'}\frac{d^2V}{dr^2}\langle n(r,\theta)\rangle + \frac{1}{V'}\oint d\theta d\phi\, n(r,\theta)\,\frac{\partial\sqrt{g}}{\partial r}\right] n_0(r)^{-1}$$

that is

$$\mathcal{G}_r = -\frac{1}{V'}\frac{d^2V}{dr^2}\,\langle \exp\{-E(r,\theta)\}\rangle + \left\langle \exp\{-E(r,\theta)\}\,\frac{\partial\log(\sqrt{g})}{\partial r}\right\rangle$$

Partial derivatives versus the minor radius r are intended to be performed at constant θ. We note that $\mathcal{G}_r = 0$ with a Hamada coordinate system (as used in GKW).

In order to derive the relationship between the two pairs (D, V) and (D_0, V_0), we proceed by expressing the flux surface averaged density $n(r)$ and its radial derivative $dn(r)/d(r)$ in terms of $n_0(r)$ and $dn_0(r)/d(r)$. The relationship between $n(r)$ and $n_0(r)$ is straightforward

$$n(r) = n_0(r)\left\langle \exp\left\{-\frac{Ze\Phi(r,\theta)}{T(r)} + \frac{m\Omega^2(r)}{2T(r)}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\}\right\rangle$$

In order to express $dn(r)/d(r)$ in terms of $n_0(r)$ and $dn_0(r)/d(r)$, we proceed with the computation of $\partial n(r,\theta)/\partial r$.

$$\begin{aligned}\frac{\partial n(r,\theta)}{\partial r} &= \exp\left\{-\frac{Ze\Phi(r,\theta)}{T(r)} + \frac{m\Omega^2(r)}{2T(r)}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\}\left\{\frac{dn_0(r)}{dr} + n_0(r)\cdot\right.\\ &\cdot\left[-\frac{Ze}{T(r)}\frac{\partial\Phi(r,\theta)}{\partial r} + \frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr}\left(R(r,\theta)^2 - R_0(r)^2\right) + \frac{m\Omega^2}{2T(r)}\frac{\partial}{\partial r}\left(R(r,\theta)^2 - R_0(r)^2\right)\right]\\ &+ \left. n_0(r)\frac{1}{T(r)}\frac{dT(r)}{dr}\left[\frac{Ze}{T(r)}\Phi(r,\theta) - \frac{m\Omega^2}{2T(r)}\left(R(r,\theta)^2 - R_0(r)^2\right)\right]\right\}\end{aligned} \tag{B.1}$$

We take now the flux surface average, and recalling that

$$\frac{dn(r)}{dr} = \left\langle\frac{\partial n(r,\theta)}{\partial r}\right\rangle + n_0(r)\,\mathcal{G}_r,$$

we find,

$$\begin{aligned}
\frac{dn(r)}{dr} &= \langle \exp\{-E(r,\theta)\}\rangle \frac{dn_0(r)}{dr} + \\
&+ n_0(r)\left\{-\left\langle \exp\{-E(r,\theta)\}\ \frac{Ze}{T(r)}\frac{\partial\Phi(r,\theta)}{\partial r}\right\rangle\right. \\
&+ \left\langle \exp\{-E(r,\theta)\}\ \left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr} \\
&+ \left\langle \exp\{-E(r,\theta)\}\ \frac{\partial}{\partial r}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)} \\
&+ \left\langle \exp\{-E(r,\theta)\}\ \frac{Ze}{T(r)}\Phi(r,\theta)\right\rangle \frac{1}{T(r)}\frac{dT(r)}{dr} \\
&- \left.\left\langle \exp\{-E(r,\theta)\}\ \left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)}\frac{1}{T(r)}\frac{dT(r)}{dr}\right\} + n_0(r)\,\mathcal{G}_r.
\end{aligned}$$

At this point, we replace the expressions of $dn(r)/dr$ and $n(r)$ in the expression for Γ and we obtain

$$\begin{aligned}
\Gamma = &-D\,\langle \exp\{-E(r,\theta)\}\rangle \frac{dn_0(r)}{dr} + \\
&-D\,n_0(r)\left\{-\left\langle \exp\{-E(r,\theta)\}\ \frac{Ze}{T(r)}\frac{\partial\Phi(r,\theta)}{\partial r}\right\rangle\right. \\
&+\left\langle \exp\{-E(r,\theta)\}\ \left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr} \\
&+\left\langle \exp\{-E(r,\theta)\}\ \frac{\partial}{\partial r}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)} \\
&+\left\langle \exp\{-E(r,\theta)\}\ \frac{Ze}{T(r)}\Phi(r,\theta)\right\rangle \frac{1}{T(r)}\frac{dT(r)}{dr} \\
&-\left.\left\langle \exp\{-E(r,\theta)\}\ \left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)}\frac{1}{T(r)}\frac{dT(r)}{dr} + \mathcal{G}_r\right\} \\
&+V\,n_0(r)\ \langle \exp\{-E(r,\theta)\}\rangle
\end{aligned}$$

Finally, we compare with equation

$$\Gamma = -D_0\frac{dn_0}{dr} + V_0 n_0,$$

and we identify the following relationships:

$$D_0 = D\,\langle \exp\{-E(r,\theta)\}\rangle\,,$$

and

$$\begin{aligned}
V_0 &= -D\Bigg\{-\left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\frac{\partial\Phi(r,\theta)}{\partial r}\right\rangle \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr} \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\frac{\partial}{\partial r}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)} \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\Phi(r,\theta)\right\rangle \frac{1}{T(r)}\frac{dT(r)}{dr} \\
&- \left\langle \exp\{-E(r,\theta)\}\,\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)}\frac{1}{T(r)}\frac{dT(r)}{dr} + \mathcal{G}_r\Bigg\} \\
&+ V\left\langle \exp\{-E(r,\theta)\}\right\rangle .
\end{aligned}$$

The final relationships follow directly

$$D = D_0\left\langle \exp\{-E(r,\theta)\}\right\rangle^{-1} = D_0\frac{n_0}{n},$$

$$\begin{aligned}
V &= V_0\left\langle \exp\{-E(r,\theta)\}\right\rangle^{-1} + D_0\Bigg\{-\left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\frac{\partial\Phi(r,\theta)}{\partial r}\right\rangle \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr} \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\frac{\partial}{\partial r}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)} \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\Phi(r,\theta)\right\rangle \frac{1}{T(r)}\frac{dT(r)}{dr} \\
&- \left\langle \exp\{-E(r,\theta)\}\,\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)}\frac{1}{T(r)}\frac{dT(r)}{dr} + \mathcal{G}_r\Bigg\}\left\langle \exp\{-E(r,\theta)\}\right\rangle^{-2}
\end{aligned}$$

We define the following conversion factor for the convection,

$$\begin{aligned}
\mathcal{V} &= a\Bigg\{-\left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\frac{\partial\Phi(r,\theta)}{\partial r}\right\rangle \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr} \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\frac{\partial}{\partial r}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)} \\
&+ \left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\Phi(r,\theta)\right\rangle \frac{1}{T(r)}\frac{dT(r)}{dr} \\
&- \left\langle \exp\{-E(r,\theta)\}\,\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle \frac{m\Omega^2}{2T(r)}\frac{1}{T(r)}\frac{dT(r)}{dr} + \mathcal{G}_r\Bigg\}\left\langle \exp\{-E(r,\theta)\}\right\rangle^{-1},
\end{aligned}$$

by which the transformation from the pair (D_0, V_0) into the pair (D, V) simply reads

$$\begin{aligned}
D &= \left\langle \exp\{-E(r,\theta)\}\right\rangle^{-1} D_0 = \frac{n_0}{n}D_0, && \text{(B.2)}\\
aV &= \left\langle \exp\{-E(r,\theta)\}\right\rangle^{-1}\left(aV_0 + D_0\mathcal{V}\right). && \text{(B.3)}
\end{aligned}$$

Here we have introduced an arbitrary normalizing length a (usually the geometrical major radius R_{geo} in GKW and the minor radius a in NEO).

The quantity $\mathcal{V}$ can be directly computed inside codes which treat poloidal aysmmetries (like GKW and NEO). It is suggested that the quantity $\mathcal{V}$ is computed for each species independently.

Thereby, for each kinetic species, codes like GKW and NEO can provide directly in output the two flux surface averaged quantities n_0/n (already present in output actually) and the convection conversion factor $\mathcal{V}$. In this way the equation to be applied in 1D transport codes to pass from (D_0, V_0) to (D, V) is very simple.

Optionally, codes like GKW and NEO can also provide in output the entire set of flux surface averaged quantities which are required to perform the transformation from the pair (D_0, V_0) into the pair (D, V),

$$
\begin{aligned}
e_0 &= \langle \exp\{-E(r,\theta)\}\rangle\,,\\
e_1 &= \left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\Phi(r,\theta)\right\rangle,\\
e_2 &= a\left\langle \exp\{-E(r,\theta)\}\,\frac{Ze}{T(r)}\frac{\partial\Phi(r,\theta)}{\partial r}\right\rangle,\\
e_3 &= a^{-2}\left\langle \exp\{-E(r,\theta)\}\,\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle,\\
e_4 &= a^{-1}\left\langle \exp\{-E(r,\theta)\}\,\frac{\partial}{\partial r}\left(R(r,\theta)^2 - R_0(r)^2\right)\right\rangle,\\
e_5 &= a\,\mathcal{G}_r = a\left[-\frac{1}{V'}\frac{d^2V}{dr^2}\,e_0 \;+\; \left\langle \exp\{-E(r,\theta)\}\,\frac{\partial\log(\sqrt{g})}{\partial r}\right\rangle\right]
\end{aligned}
$$

where we included normalizations to an arbitrary length a, which can be chosen in order to be consistent with the other normalizations used in the codes (that is, e.g., major radius in GKW, and minor radius in NEO). We remind that $e_5 = 0$ with Hamada coordinates, as used in GKW. Then, the convection conversion factor reads

$$
\mathcal{V} = \frac{1}{e_0}\left\{-e_2 + e_3 a^3\frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr} + e_4 a^2\frac{m\Omega^2}{2T(r)} + e_1 a\frac{1}{T(r)}\frac{dT(r)}{dr} - e_3 a^3\frac{m\Omega^2}{2T(r)}\frac{1}{T(r)}\frac{dT(r)}{dr} + e_5\right\}.
$$

and the transformation equations read

$$
\begin{aligned}
D &= \frac{D_0}{e_0} = \frac{n_0}{n}D_0,\\
aV &= (aV_0 \;+\; D_0\mathcal{V})\,{e_0}^{-1}.
\end{aligned}
$$

We notice that the convection conversion factor $\mathcal{V}$ has been defined in such a way that

$$
\frac{aV}{D} = \frac{aV_0}{D_0} + \mathcal{V},
$$

which can also be regarded as a direct consequence of the following relationship

$$
\frac{a}{n}\frac{dn}{dr} = \frac{a}{n_0}\frac{dn_0}{dr} + \mathcal{V}.
$$

B.0.1 Anisotropic minorities (Reinke model)

In the Reinke model (see Sec. 4.4.1), the poloidal density variation for the anisotropic minority is

$$
n(r,\theta) = n_0\left(\frac{B}{B_0}\right)^{-\eta}\exp(-\mathcal{E}_\Omega/T_\parallel) = n_0\exp(-\mathcal{E}_{\Omega,\eta}/T_\parallel) \tag{B.4}
$$

The radial deriviative then has additional terms

$$\frac{\partial n(r,\theta)}{\partial r} = n_0 \left[-\ln(B/B_0)\frac{\partial \eta}{\partial r} - \frac{\eta}{B}\frac{\partial B}{\partial r} + \frac{\eta}{B_0}\frac{\partial B_0}{\partial r} \right] \exp(-\mathcal{E}_{\Omega,\eta}/T_\parallel) + \cdots \tag{B.5}$$

where $\cdots$ represents all the terms on the RHS of equation B.1 with the generalised normalised energy $E = \mathcal{E}_{\Omega,\eta}/T_\parallel$ in place of the former $E = \mathcal{E}_\Omega/T$ (as defined in Eq. 4.36). The coefficients $e_0, e_1, e_2, e_3, e_4, e_5$ are also generalised using this generalised energy in their definition. Then, the transformation identities above all still hold if $\mathcal{V}$ is extended to

$$\mathcal{V} = \frac{1}{e_0}\left\{ -e_2 + e_3 a^3 \frac{m\Omega}{T(r)}\frac{d\Omega(r)}{dr} + e_4 a^2 \frac{m\Omega^2}{2T(r)} + e_1 a \frac{1}{T(r)}\frac{dT(r)}{dr} - e_3 a^3 \frac{m\Omega^2}{2T(r)}\frac{1}{T(r)}\frac{dT(r)}{dr} + e_5 + e_\eta \right\}. \tag{B.6}$$

where the additional coefficient e_η is defined as

$$e_\eta = a\left\langle \exp\left\{-\mathcal{E}_{\Omega,\eta}(r,\theta)/T_\parallel\right\} \left[-\ln(B/B_0)\frac{\partial \eta}{\partial r} - \frac{\eta}{B}\frac{\partial B}{\partial r} + \frac{\eta}{B_0}\frac{\partial B_0}{\partial r} \right] \right\rangle .$$

.

B.0.2 Anisotropic minorities (Bilato model)

In the Bilato model (see Sec. 4.4.2), the poloidal density variation for the anisotropic minority is

$$n_m = n_0 \frac{T_\perp(\theta)}{T_{\perp 0}} \exp(-\mathcal{E}_\Omega/T_\parallel) = n_0 \exp(-\mathcal{E}_{\Omega,\eta}/T_\parallel) \tag{B.7}$$

where

$$\frac{T_\perp(\theta)}{T_{\perp 0}} = \left[\frac{T_{\perp 0}}{T_\parallel} + \left(1 - \frac{T_{\perp 0}}{T_\parallel}\right)\frac{B_0}{B} \right]^{-1} . \tag{B.8}$$

The radial deriviative then has additional terms

$$\frac{\partial n(r,\theta)}{\partial r} = -n_0 \left[\frac{\partial (T_{\perp 0}/T_\parallel)}{\partial r}\left(1 - \frac{B_0}{B}\right) + \left(-\frac{B_0}{B^2}\frac{\partial B}{\partial r} + \frac{1}{B}\frac{\partial B_0}{\partial r} \right)\left(1 - \frac{T_{\perp 0}}{T_\parallel}\right) \right] \frac{T_\perp(\theta)}{T_{\perp 0}} \exp(-\mathcal{E}_{\Omega,\eta}/T_\parallel) + \cdots \tag{B.9}$$

where $\cdots$ represents all the terms on the RHS of equation B.1 with the generalised normalised energy $E = \mathcal{E}_{\Omega,\eta}/T_\parallel$ in place of the former $E = \mathcal{E}_\Omega/T$ (as defined in Eq. 4.41). The coefficients $e_0, e_1, e_2, e_3, e_4, e_5$ are also generalised using this generalised energy in their definition. Then $\mathcal{V}$ is extended as in Eq. B.6 with the additional coefficient e_η is defined as

$$e_\eta = -a\left\langle \exp\left\{-\mathcal{E}_{\Omega,\eta}(r,\theta)/T_\parallel\right\} \left[\frac{\partial (T_{\perp 0}/T_\parallel)}{\partial r}\left(1 - \frac{B_0}{B}\right) + \left(-\frac{B_0}{B^2}\frac{\partial B}{\partial r} + \frac{1}{B}\frac{\partial B_0}{\partial r} \right)\left(1 - \frac{T_{\perp 0}}{T_\parallel}\right) \right] \frac{T_\perp(\theta)}{T_{\perp 0}} \right\rangle .$$

Bibliography

[1] A.G. Peeters, Y. Camenen, F.J. Casson, W.A. Hornsby, A.P. Snodin, D. Strintzi, G. Szepesi, Computer Physics Communications, **180**, 2650, (2009). doi: 10.1016/j.cpc.2009.07.001.

[2] A.G. Peeters, D. Strintzi, Phys. Plasmas **11** (2004) 3748

[3] A.G. Peeters, C. Angioni, D. Strintzi, Phys. Rev. Lett. **98** (2007) 265003

[4] S.E. Parker, W.W. Lee, R.A. Sandtoro, Phys. Rev. Lett. **71** (1993) 2042

[5] Z. Lin, T.S. Hahm, W.W. Lee, et al., Science **281** (1998) 1835

[6] Y. Idomura, S. Tokuda, Y. Kishimoto, Nucl. Fusion **43** (2003) 234

[7] J.A. Heikkinen, J.S. Janhunen, T.P. Kiviniemi, et al., Contr. Plas. Phys. **44** (2004) 13

[8] A. Bottino, A.G. Peeters, R. Hatzky, et al., Phys. Plasmas **14** (2007) 010701

[9] S. Jolliet, A. Bottino, P. Angelino, et al., Comp. Phys. Comm. **177** (2007) 409

[10] V. Grandgirard, M. Brunetti, P. Bertrand, et al., J. Comp. Physics **217** (2006) 395

[11] M. Kotschenreuther, G. Rewoldt, W.M. Tang., Comp. Phys. Comm. **88** (1995) 128

[12] W. Dorland, F. Jenko, M. Kotschenreuther, et al., Phys. Rev. Lett. **85** (2000) 5579

[13] F. Jenko, Comp. Phys. Comm. **125** (2000) 196

[14] T. Dannert, F. Jenko, Phys Plasmas **12** (2005) 072309

[15] J. Candy, R.E. Waltz, J. Comp. Phys. **186** (2003) 545

[16] B.D. Scott *delta-FEFI*

[17] A.G. Peeters, C. Angioni, Phys. Plasmas **12** (2005) 072515

[18] A.G. Peeters, C. Angioni, A. Bottino, et al. Plasma Phys. Contr. Fusion **48** (2006) B413

[19] A.G. Peeters, C. Angioni, D. Strintzi, Phys. Plasmas **16** (2009) 034703

[20] A.G. Peeters, D. Strintzi, C. Angioni, et al., Phys. Plasmas **16** (2009) 042310 (Note also the Erratum to this paper)

[21] Y. Camenen, A.G. Peeters, C. Angioni, Phys. Rev. Lett. **102** (2009) 125001

[22] A.G. Peeters, C.Angioni, Y. Camenen, et al., Phys. Plasmas. **16** (2009) 062311

[23] Y. Camenen, A.G. Peeters, C. Angioni, et al., Phys. Plasmas **16** (2009) 062501

[24] F.J. Casson, A.G. Peeters, Y. Camenen, et al., Phys. Plasmas **16** (2009) 092303

[25] Y. Camenen, A.G. Peeters, C. Angioni, et al., Phys. Plasmas **16** (2009) 012503

[26] F.J. Casson, A.G. Peeters, C. Angioni, et al., Phys. Plasmas **17** (2010) 102305

[27] W.A. Hornsby, A.G. Peeters, A.P. Snodin, et. al., Phys. Plasmas **17** (2010) 092301

[28] W.A. Hornsby, A.G. Peeters, E. Poli, et al., Euro. Phys. Let. **91** (2010) 45001

[29] R.G. Littlejohn, J. Plasma Phys. **29** (1983) 111

[30] T.S. Hahm, Phys. Fluids **31** (1988) 043207

[31] A. Brizard, Phys. Plasmas **41** (1988) 541

[32] H. Sugama, Phys. Plasmas **7** (2000) 466

[33] A. Brizard, Rev. Modern Phys.**79** (2007) 421

[34] M. Beer, PhD thesis, Princeton University 1995

[35] A. J. Brizard, Phys. Plasmas **2** (1995) 459

[36] S. Hamada, Kakuyugo Kenkyu **1** (1958) 542

[37] R. L. Miller, M. S. Chu, J. M. Greene, Y. R. Lin-Liu, R. E. Waltz, Phys. Plasmas **5** (1998) 973

[38] J.Candy, Plasma Phys. Control. Fusion **51** (2009) 105009

[39] A.Thyagaraja, K.G.McClements, Phys. Plasmas 13, 062502 (2006)

[40] B. D. Scott, Phys Plasmas **5** (1998) 2334

[41] B. D. Scott, Phys Plasmas **8** (2001) 447

[42] H. Lütjens, A. Bondeson and O. Sauter, Comput. Phys. Commun.**97** (1996) 219

[43] J. W. Connor, R . J. Hastie and J. B. Taylor, Phys. Rev. Let., **40** (1978) 396

[44] X. Lapillonne, S. Brunner, T. Dannert, S. Jolliet, A. Marinoni, L. Villard, T. Goerler, F. Jenko, and F. Merz, Phys. Plasmas, **16**, 032308, (2009).

[45] W. Dorland, M. Kotschenreuther, *Notes on Local Equilibrium Implementation* available from http://w3.pppl.gov/ hammett/work/gs2/docs/geometry/g_short.pdf

[46] C. F. F. Karney, Comp. Phys. Reports **4** (1986) 183

[47] C. Veth, C. Angioni, 2011, *Report on grid convergence with GKW* available from http://www.gkw.org.uk/tikiwiki/Manual

[48] A.M. Dimits, Phys. Plasmas **7** (2000) 969

[49] G. L. Falchetto, B. D Scott, P. Angelino, A. Bottino, T. Dannert, V. Grandgirard, S. Janhunen, F. Jenko, S. Jolliet, A. Kendl, B. F. McMillan, V. Naulin, A. H. Nielsen, M. Ottaviani, A. G. Peeters, M. J. Pueschel, D. Reiser, T. T. Ribeiro, and M. Romanelli, Plasma Physics and Controlled Fusion **50** (2008) 124015

[50] C. M. Roach, et. Al. Plasma Phys. Control. Fusion **51**, 124020, (2009).

[51] F. J. Casson, PhD Thesis, University of Warwick, 2011. http://wrap.warwick.ac.uk/36765/

[52] F. J. Casson, 2011, *GKW benchmark of background ExB shear with GYRO* available from http://www.gkw.org.uk/tikiwiki/Manual

[53] J. E. Kinsey, R. E. Waltz, and J. Candy, Phys. Plasmas **12**, 062302 (2005).

[54] Fable, E., C. Angioni, F. J. Casson, D. Told, et al., Plasma Physics and Controlled Fusion **55**, 124028 (2013)

[55] D. Told, F. Jenko, T. Goerler, F. J. Casson, E. Fable, and ASDEX Upgrade Team., Phys. Plasmas **20**, 122312 (2013)

[56] M. Frigo, S. G. Johnson, Proc. IEEE **93** (2005) 216

[57] C. Campos, J. E. Roman, E. Romero and A. Tomas, SLEPc Users Manual **DSIC-II/24/02 - Revision 3.3**, D. Sistemes Informàtics i Computació, Universitat Politècnica de València, 2012. http://www.grycap.upv.es/slepc/documentation/manual.htm

[58] T. S. Hahm and K. H. Burrell. *Phys. Plasmas*, **2** (1995) 1648

[59] K. H. Burrell. Phys. Plasmas, **4** (1997) 1499

[60] R. E. Waltz, G. D. Kerbel, and J. Milovich, Phys. Plasmas **1** (1994) 2229

[61] R. L. Miller and R. E. Waltz, Phys. Plasmas **1** (1994) 2835

[62] F. Baron. PhD in physics, (Univ. Pierre et Marie Curie, Paris, France, 1982)

[63] U. Schumann, Algorithms for direct numerical simulation of shear-periodic turbulence, in *Lecture Notes in Physics* **218** (Berlin Springer Verlag 1985) pages 492

[64] T. Gerz, U. Schumann, and S. E. Elghobashi, Journal of Fluid Mechanics Digital Archive **200** (1989) 563

[65] R. S. Rogallo, *Numerical experiments in homogeneous turbulence. NASA STI/Recon Technical Report N* (1981) 81:31508

[66] T. A. Zang, Applied Numerical Mathematics **7** (1991) 27

[67] Alain Pumir, Phys. Fluids, **8** (1996) 3112

[68] G. W. Hammett, W. Dorland, N. F. Loureiro, and T. Tatsuno. Implementation of Large Scale E x B Shear Flow in the GS2 Gyrokinetic Turbulence Code. *APS Meeting Abstracts* (2006) pages 1136

[69] F. Jenko, T. Dannert, C. Angioni, Plasma Phys. Contr. Fusion **47** (2005) B195

[70] W.W. Lee, J.L.V. Lewandowski, T.S. Hahm, and Z. Lin, Phys. Plasmas **8**, 4435 (2001)

[71] F. Merz and F. Jenko, Nucl. Fusion **50**, 054005 (2010)

[72] F. L. Hinton, M. N. Rosenbluth, Phys. Rev. Lett. **80** (1998) 724

[73] Y. Xiao, P. J. Catto, Phys. Plasmas **13** (2006) 102311

[74] A. G. Peeters, C. Angioni, M. Apostoliceanu et al., Phys. Plasma **12** (2005) 022505

[75] W. Nevins, et al., Phys. Plasmas **13** (2006) 122306

[76] A. Arakawa, J. Comp. Phys. **1** (1966) 119

[77] J.Dunkel, P.Talkner, B. Hänggi, New Journal of Physics **9** (2007) 114

[78] M. L. Reinke et al, Plasma Phys. Contr. Fusion **54** (2012) 045004

[79] R. Bilato, O. Maj, C. Angioni, Nucl. Fusion (letter), **56** (2014)

[80] T. Sung, R. Buchholz, F.J. Casson, E. Fable, S.R. Grosshauser, W.A. Hornsby, P. Migliano and A.G. Peeters, Phys. Plasmas **20**, 042506 (2013)

Index

Nomenclature

H	The Hamiltonian
L	The Lagrangian
L_p	The particle Lagrangian
T	The temperature
$\mathbf{X}$	The gyro-centre spatial coordinate
Z	The charge number
e	The unit charge
f	The perturbed distribution
F_M	The Maxwell distribution
$\mathbf{v}$	The velocity
ϕ	The perturbed potential
$\langle\rangle$	The gyro-average operation
$\langle\rangle^\dagger$	The conjugate transpose of the gyro-average operator