HMGC User guide

Last update: June 13, 2008

G. Vlad, S. Briguglio, G. Fogaccia, C. Di Troia

Associazione Euratom-ENEA sulla Fusione, C.R. Frascati,
C.P. 65 - I-00044 - Frascati, Rome, Italy

Manuale utente di HMGC

G. Vlad, S. Briguglio, G. Fogaccia, C. Di Troia

Riassunto

Questo manuale descrive l'utilizzo di HMGC (Hybrid MHD Gyrokinetic Code), il codice di simulazione 3-D ibrido magnetoidrodinamico-girocinetico a particelle, sviluppato a Frascati nei primi anni '90. HMGC è stato sviluppato per studiare l'interazione nonlineare di ioni energetici con turbolenza di tipo Alfvénico, in plasmi che bruciano. Il modello di plasma adottato nel codice HMGC consiste in una componente di plasma termico (core) e una popolazione di ioni energetici. La prima è descritta dalle equazioni della Magneto-idro-dinamica (MHD) ridotte $O(\epsilon^3)$ nel limite di pressione nulla (dove ϵ è l'inverso del rapporto di aspetto del toro), inclusi termini resistivi e viscosi. La popolazione di ioni energetici è descritta dall'equazione di Vlasov girocinetica nonlineare, espansa nel limite $k_\perp \rho_H \ll 1$ ($k_\perp$ essendo la componente perpendicolare al campo magnetico del vettore d'onda, e ρ_H il raggio di Larmor degli ioni energetici), con gli effetti di orbita di deriva magnetica pienamente ritenuti, e risolta con tecniche particle-in-cell (PIC). Lo scopo di questo manuale utente è quello di rendere il lettore in grado di utilizzare il codice e di analizzarne i risultati con un insieme di strumenti grafici, anch'essi descritti con un certo dettaglio.

Parole chiave: Prodotti di fusione, Particelle alfa, particelle veloci, Magnetoidrodinamica (MHD), Onde di Alfvén, Tokamaks, Tecniche Particle-in-cell (PIC), Simulazioni girocinetiche

HMGC user manual

Abstract

This user guide describes the use of HMGC (Hybrid MHD Gyrokinetic Code), the hybrid MHD-particle 3-D simulation code developed in Frascati in the early 90s. HMGC has been written in order to study nonlinear interactions of energetic ions with the Alfvénic turbulence in burning plasma conditions. The plasma model adopted in the HMGC code consists of a thermal (core) plasma and an energetic-ion population. The former is described by reduced $O(\epsilon^3)$ Magneto-Hydro-Dynamics (MHD) equations in the limit of zero pressure (ϵ being the inverse aspect ratio of the torus), including resistivity and viscosity terms. The energetic-ion population is described by the nonlinear gyrokinetic Vlasov equation, expanded in the limit $k_\perp \rho_H \ll 1$ (with $k_\perp$ being the perpendicular component of the wave vector to the magnetic field, and ρ_H the energetic-ion Larmor radius), though fully retaining magnetic drift orbit widths, and solved by particle-in-cell (PIC) techniques. The aim of this user guide is to make the reader able to run the code and analyze its results using a suite of graphics tools, also described in some detail.

Keywords: *Fusion products, Alpha particles, Fast particles, Magnetohydrodynamic (MHD), Alfvén waves, Tokamaks, Particle-in-cell (PIC) techniques, Gyrokinetic simulations*

Contents

List of Figures

List of Tables

HMGC User guide

1 Introduction

In a burning plasma, fast ions (MeV energies) produced by additional heating and/or fusion reactions are expected to transfer their energy via Coulomb collisions to the thermal plasma particles (10keV energies). Due to their high velocity, of the order of Alfvén velocity, they can resonate with and possibly destabilize Alfvén modes. Energetic ion transport and confinement properties – of crucial importance for achieving efficient plasma heating and, therefore, ignition conditions – can in turn be affected by nonlinear interactions with the Alfvénic turbulence. Thus large efforts have been devoted to assess the stability of shear-Alfvén modes in tokamaks and to investigate their effect on the energetic ion transport.

The need of fully retaining nonlinear dynamics and properly taking into account kinetic effects, such as resonant interactions between energetic ions and Alfvén modes and the nonperturbative character of such interactions makes the numerical particle-simulation approach the natural tool for this investigation.

A hybrid MHD-particle 3-D simulation code, HMGC (Hybrid MHD Gyrokinetic Code) has been developed in Frascati in the early 90's.

The plasma model adopted in the HMGC code consists of a thermal (core) plasma and an energetic-ion population. The former is described by reduced $O(\epsilon^3)$ Magneto-Hydro-Dynamics (MHD) equations in the limit of zero pressure (ϵ being the inverse aspect ratio of the torus), including resistivity and viscosity terms. The reduced MHD equations expanded to $O(\epsilon^3)$ allow us to investigate equilibria with shifted circular magnetic surfaces. The energetic-ion population is described by the nonlinear gyrokinetic Vlasov equation [1, 2], expanded in the limit $k_\perp \rho_H \ll 1$ (with $k_\perp$ being the perpendicular component of the wave vector to the magnetic field, and ρ_H the energetic-ion Larmor radius), though fully retaining magnetic drift orbit widths, and solved by particle-in-cell (PIC) techniques. The coupling between energetic ions and thermal plasma is obtained through the divergence of the energetic-ion pressure tensor, which enters the vorticity equation. Numerical simulations of experimental conditions are performed by fitting the relevant thermal-plasma quantities – the on-axis equilibrium magnetic field, major and minor radii (R_0 and a, respectively), the safety-factor q, the electron n_e and ion n_i plasma densities, the electron temperature T_e –, the anisotropy of the energetic-ion distribution function and the ratio β_H between fast ion and magnetic pressures.

In order to retain the relevant finite Larmor radius effects without resolving the details of the gyromotion, the energetic ions are evolved in their gyrocenter coordinate system, which corresponds to averaging the single-particle equations of motion over the fast Larmor precession.

HMGC has been successfully applied to the interpretation of the experimental evidences of rapid transport of energetic ions related with fluctuations in the Alfvén-mode frequency range in auxiliary-heated JT-60U discharges, in connection with so called Abrupt Large-amplitude Events (ALEs) [3, 4, 5]. HMGC results have also suggested a possible justification of the large discrepancy, observed in reversed-shear beam-heated DIII-D discharges, between the energetic particle radial density profile expected from classical deposition and that deduced from the experimental measurements.

In spite of the slightly simplified physical model, HMGC has been getting increasing attention from the international plasma physics community, and it has been recently acquired by EPFL CRPP Lausanne, University of California Irvine and IFTS Zhejiang University.

Aim of the present report is yielding a HMGC User Guide. We proceed with a summary description of the various sections. In Sect. (2) it is shown how to produce a plasma equilibrium needed by HMGC. In Sect. (3) it is described the execution script which prepares the set of input files required for compilation and execution of the code. In particular, the script prepares both the sets of files required by the two modules that constitute the hybrid code: the MHD module and the gyrokinetic one. Sects. (4) and (5) describe the output files of the MHD part and of the gyrokinetic one, respectively; they also describe the suite of graphics tools used to post-process and visualize the results contained in these files. Sect. (6) describes the three types of energetic particles distribution functions that can be loaded to start a simulation: the slowing down, the maxwellian and the bi-maxwellian distribution functions. The various operations needed to setup a run of HMGC have been collected in Sect. (7), where a specific HMGC run referred to a DIII-D, beam-heated discharge is used as an example. Sect. (8) shows the list of the directories tree structure of HMGC. Finally, in Sect. (9) several excerpts of Ref. [9] are reported to illustrate the analytical details of the model that constitutes the basis of HMGC.

2 How to produce an MHD equilibrium file

The equilibrium file required by `HMGC` is produced by running the fortran file `eqe3aaab`. This program solves the Grad-Shafranov equation expanded to the $O(\epsilon^3)$ in the inverse aspect ratio $\epsilon \equiv a/R_0$, with a and R_0 the minor and major radius of the torus, respectively for the poloidal flux function ψ (see Sect.(9)), assuming an analytic parametrization of the safety factor profile $q = q(r)$ (with r the normalized minor radius $r \equiv r/a$, a being the minor radius of the circular cross section torus) given by:

$$q(r) \;=\; q_0 \left[1 + \left(\frac{r}{r_0} \right)^{2\lambda} \right]^{1/\lambda} , \tag{1}$$

with r_0 defined in terms of λ and the q value at the centre $q(r = 0) \equiv q_0$ and at the edge $q(r = 1) \equiv q_a$:

$$r_0 \;=\; \left[\left(\frac{q_a}{q_0} \right)^{\lambda} - 1 \right]^{-1/2\lambda} . \tag{2}$$

The normalized (to B_T/R_0) current density profile and the shear profile can be derived from the previous expressions:

$$j(r) \;=\; \frac{2}{q_0 \left[1 + \left(\frac{r}{r_0} \right)^{2\lambda} \right]^{\frac{1}{\lambda}+1}} , \tag{3}$$

$$\hat{s}(r) \;=\; \left(\frac{r}{r_0} \right)^{2\lambda} \frac{2}{\left[1 + \left(\frac{r}{r_0} \right)^{2\lambda} \right]} , \tag{4}$$

From the above expressions the normalized (to $B_T a^2/R_0$) Fourier components $\psi_{m,0}$ for the equilibrium poloidal flux function are obtained, namely $\psi_{0,0}, \psi_{1,0}$ (here m is the poloidal mode number, and the toroidal mode number $n = 0$ has been assumed for the axisymmetric equilibrium). **Please note that the normalizations in the gyrokinetic module will be different**.

The expressions shown in eqs. (1), (2) (3) (4) are appropriate for describing a monotonic q-profile, but they are inadequate to describe more general safety factor profiles, as, *e.g.*, reversed shear profiles. Thus a number of *bumps* on the current density profile can be superimposed on eq. (3). Actually up to 3 bumps can be superimposed:

$$j(r) \quad = \quad \frac{2}{q_0 \left[1 + \left(\frac{r}{r_0}\right)^{2\lambda}\right]^{\frac{1}{\lambda}+1}} + \sum_{i=1,3} \mathrm{bump}_{\mathrm{eq,i}} e^{-\left(\frac{r^2 - cg_i}{wg_i}\right)^2} , \tag{5}$$

where $\mathrm{bump}_{\mathrm{eq,i}}$ can be positive or negative. The current density profile resulting from Eq.(5) is then rescaled and such to provide a q profile which has the minimum equal to the parameter q_0 of Eq. (1).

The meaning of the different parameters of the input file (`EQUIPA`) (assigned as a `namelist` with the same name of the input file) are listed in Table (1).

`Q0`	minimum q value
`Q1`	maximum value of q at $r/a = 1$: q_a (if $\mathrm{bump}_{\mathrm{eq,i}} = 0$)
`RL`	λ
`NREQ`	Number of points in the radial mesh
`NMESHA`	parameter of non equally spaced mesh (usually not used)
`NPOIDA`	parameter of non equally spaced mesh (usually not used)
`SOLPDA`	parameter of non equally spaced mesh (usually not used)
`APLACE(i)`	parameters of non equally spaced mesh (usually not used)
`AWIDTH(i)`	parameters of non equally spaced mesh (usually not used)
`EPSILO`	inverse aspect ratio ($\epsilon \equiv a/R_0$)
`RHOFLG`	logical value, if `.true.` compute $\eta(r)/\eta_0 = j_0/j(r)$
`BETA0`	parameter for equilibrium pressure profile (usually not used)
`C1, C2, C3, C4, C5`	parameters for equilibrium pressure profile (usually not used)
`BUMPEQ`, `CG`, `WG`	$\mathrm{bump}_{\mathrm{eq,1}}$, cg_1, wg_1
`BUMPEQ1`, `CG1`, `WG1`	$\mathrm{bump}_{\mathrm{eq,2}}$, cg_2, wg_2
`BUMPEQ2`, `CG2`, `WG2`	$\mathrm{bump}_{\mathrm{eq,3}}$, cg_3, wg_3
`ireadcur`	parameter to read current density profile as alternate input

Table 1: Parameters in the file `EQUIPA`.

To help in fitting an experimental q-profile, an utility to compare the experimental profile with the one obtained with the program `eqe3aaab` is provided (program `plot_equil`, see fig. (1)). In figure (1) the effect of including or not including the bumps in the current profile is shown (`&EQUIPA Q0 = 2.4110D0, Q1 = 5.1280D0, RL = 4.0D0, EPSILO = 0.293217665d0, BUMPEQ = 1.30D0, CG = 0.4000D0, WG = 0.350D0, bumpeq1 = -2.20D0, cg1 = 0.90D0, wg1 = 0.30D0, &END`). Please note that peculiar shaping of the current density profile should be avoided as much as possible, in order to prevent the (not desirable) growth of MHD unstable modes. Note that a positive bump in the current profile is used to produce an off-axis minimum in the q-profile, whereas a positive bump at the edge is used to "pull-up" the q-profile at the edge.

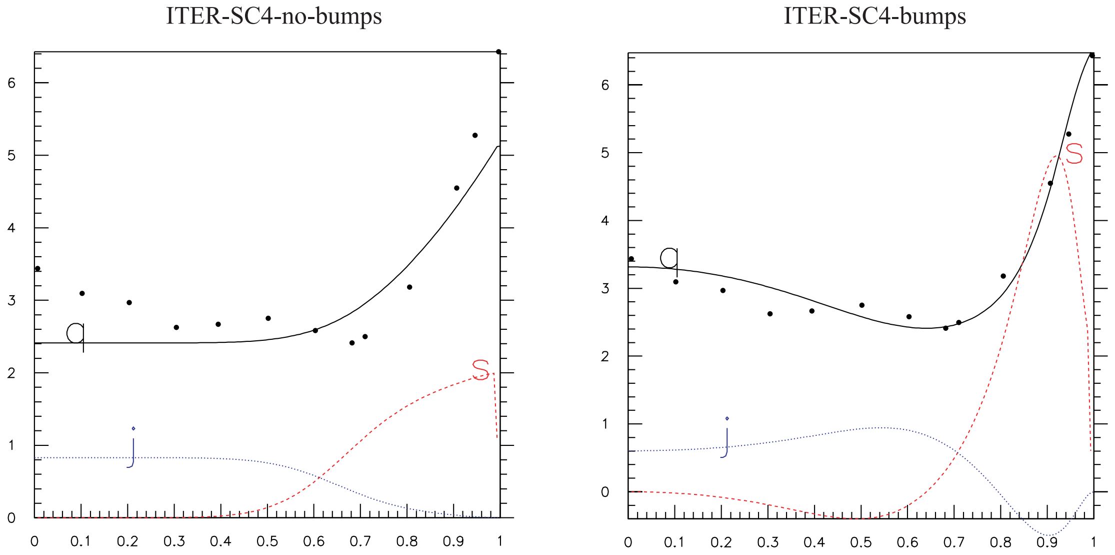

Figure 1: ITER-SC4 q-profile example of use of current bumps: (left) no bumps, (right) with two bumps (one bump is positive in amplitude (`BUMPEQ = 1.30D0`) located at r^2= `CG = 0.4000D0` ($r \simeq 0.632$) having width `WG = 0.350D0` and the second is negative (`bumpeq1 = -2.20D0`) at r^2= `cg1 = 0.90D0` ($r \simeq 0.949$) having width `wg1 = 0.30D0`).

In figure (2) the q profile used to simulate the DIII-D discharge #122117 at $t = 0.414$ s is shown. Hereafter it follows the `EQUIPA` namelist used:

. . .

```
&EQUIPA
       Q0 = 3.9891D0,
       Q1 = 15.000D0,
      RL  =  2.5D0,
      NREQ=  150,
   NMESHA =    0,
   NPOIDA =    2,
APLACE(1) = 0.426D0, 0.9D0, 0.00D0, 0., 0., 0., 0., 0., 0., 0.,
AWIDTH(1) = 0.100D0, 0.10D0, 0.00D0, 0., 0., 0., 0., 0., 0., 0.,
   SOLPDA = 0.60D0,
    EPSILO= 0.360781991d0,
    RHOFLG=.FALSE.,
    BETA0 = 0.00000D-0,
    C1    = -1.7438D0,
    C2    = -2.3515D0,
    C3    = 12.01D0,
    C4    =-15.988D0,
    C5    = 7.3964D0,
    BUMPEQ= 0.75D0,
    CG    = 0.2000D0,
    WG    = 0.220D0,
    bumpeq1 =-0.00D0,
    cg1     = 0.90D0,
    wg1     = 0.30D0,
    bumpeq2 =-0.00D0,
    cg2     = 0.95D0,
    wg2     = 0.20d0,
    ireadcur= 0,
&END
```

. . .

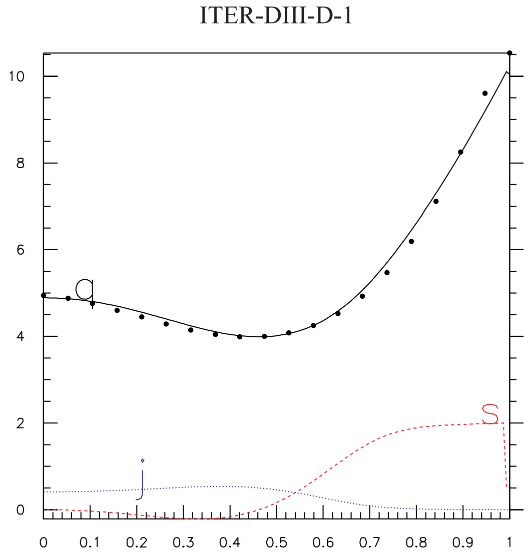

Figure 2: DIII-D discharge #122117 at $t = 0.414$ s. The parameter used are Q0 = 3.9891D0, Q1 = 15.000D0, RL = 2.5D0, NREQ= 150, EPSILO= 0.360781991d0, BUMPEQ= 0.75D0, CG = 0.2000D0, WG = 0.220D0.

The output of the program eqe3aaab is a file named EQNEW. This file will be copied by the HMGC execution script to the file named INCOND. Its structure is shown in table (2).

Quantities	Comments
the `EQUIPA namelist`	see Table (1)
`0.D0`	
`NR, 1, 0`	`FORMAT(3I20)`, `NR` is the number of radial grid points
`R(I), I=1,NR`	`FORMAT(2D30.15)` the normalized radial coordinate
	a sequence of radial profiles for the $(m = 0, n = 0)$ and $(m = 1, n = 0)$ Fourier components for ψ, ϕ and resistivity profile η in the following format:
two blank lines	
`PSI 1`	a line containing the following text: `PSI 1` for $\psi_{m,n}(r)$, `PHI 3` for $\phi_{m,n}(r)$ or `RES 4` for $\eta_{m,n}(r)$
real(m), real(n)	`FORMAT(2F20.0)`, m, n being the poloidal and toroidal mode numbers, respectively (for the equilibrium is $n = 0$)
real($\psi_{m,n}$(I)), imag($\psi_{m,n}$(I))	`FORMAT(2E30.15)`, `I=1,NR+1` (only NR points for ϕ and η)

Table 2: Structure of the file `EQNEW`.

Note that here $n = 0$ (equilibrium); also note that the $\psi_{m,n}(r)$ harmonics have one more radial point (`NR+1`) corresponding to the position of a resistive wall (this option is usually not considered). The electrostatic scalar potential $\phi_{m,n}(r)$ components for the equilibrium are identically zero (equilibrium without fluid flow), and usually (MHD module of `HMGC` used in linear mode) the resistivity profile is taken constant in radius ($\eta_{0,0} = 1$, $\eta_{1,0} = 0$). Note also that `HMGC` defines the $\psi_{0,0}$ to be $\psi_{0,0}(r = 1) = 0$ and positive in the plasma.

3 Execution script

The execution script of HMGC prepares a number of files used for compiling and running HMCG. Hereafter is a list of them, referring to a DIII-D case (see fig. (2)) simulation which consider an equilibrium with $m = 0, 1$ and $n = 0$ modes (this is mandatory) and a perturbed $n = 2$ mode, with poloidal components ranging from $m = 1$ to $m = 21$. Note that because of symmetry conditions in the Fourier space, only modes in a half (m, n) plane are required, the other ones being considered using the rule $\psi_{m,n}(r) = \psi^*_{-m,-n}(r)$ (reality of $\psi(r, \theta, \varphi)$). The choice of considering only the mode in the positive half plane defined by $n \geq 0$ is used. More over, the conventions for the Fourier transform are:

$$\begin{aligned} \psi(r,\theta,\varphi) &= \psi_{0,0}(r) + \\ &\quad 2 \sum_{l=2,LM} \left[Re(\psi_{m,n}(r)) \cos(m\theta - n\varphi) - Im(\psi_{m,n}(r)) \sin(m\theta - n\varphi) \right] , \quad (6) \\ \psi_{m,n}(r) &= \frac{1}{N_\theta N_\varphi} \sum_{j=1,N_\theta} \sum_{k=1,N_\varphi} e^{-i(m\theta_j - n\varphi_k)} \psi(r, \theta_j, \varphi_k) \, , \quad (7) \end{aligned}$$

with l being the mode index, $m = m(l)$, $n = n(l)$, LM the total number of Fourier components in the simulation (see Sect. (3.4)), N_θ and N_φ the mesh points of the θ and φ grids, respectively.

The choice for the poloidal Fourier components included in the simulation derives usually from considering $m_{min} \approx nq_{min}$, $m_{max} \approx nq_{max}$. Some restrictions could be imposed by FFT requirements (see Sect. (3.1)).

3.1 Include file `modi_inc`

Parameter definitions for compiling the MHD module (`e3_complete.F`) of `HMGC`. `NR` is the MHD radial grid (must be `NR=NREQ`, see Table (1)). `LM` is the number of Fourier components considered in the simulations. `NMAX`$=n_{max}+1$ is the maximum toroidal mode included in the simulation n plus unity. `MMAX` is the maximum number of poloidal Fourier components for fixed toroidal mode number n. `MAXPRI` is a parameter to dimension a buffer for certain output quantities. In Fig. (3) are shown two sketches of the (m, n) plane used by `HMGC`, for better clarify the parameters meaning. A constraint given by FFT routines impose that `4*(MMAX-1)` is a valid number for the FFT (see, e.g., http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.essl43.guideref.doc/am501_formul2.html). Actually, the IBM ESSL package is used, but routines which use NAG modules are also included in the source files (although they could be out of date).

```
...
      PARAMETER          (NR=150,NMAX=3,MMAX=21,LM=23,MAXPRI=200)
...
```

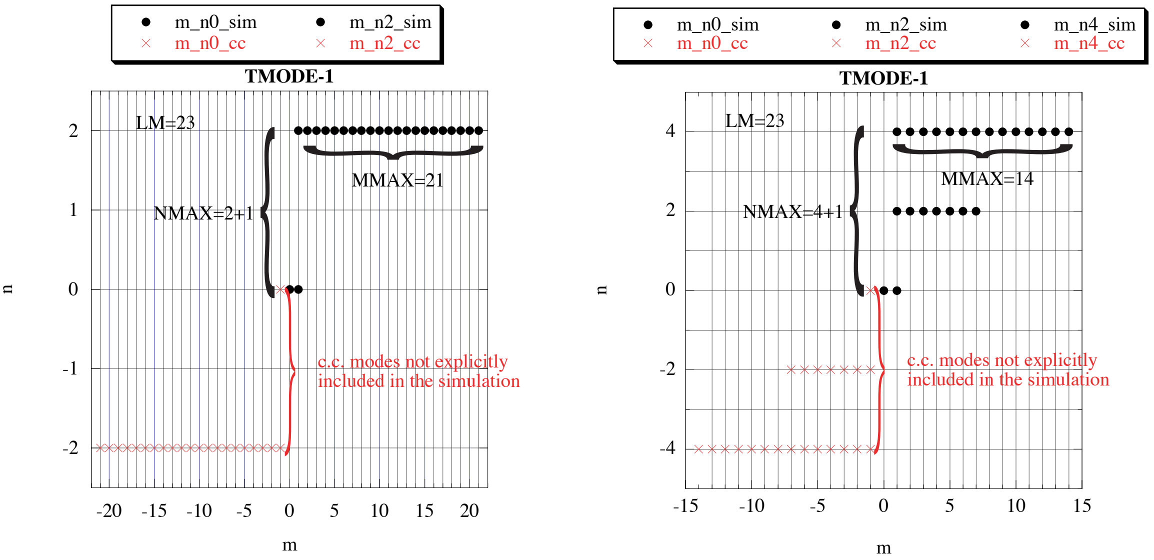

Figure 3: Fourier modes included in a `HMGC` simulation: (left) the $n = 2$ reference DIII-D case, (right) a $n = 2, 4$ case. Black dots represent the modes actually included in the simulations, the red crosses represent the modes considered in the simulation because of complex conjugate condition.

3.2 Input file PARAM

This input file contains the main input parameters for the MHD module, and for some general input parameters for the simulation. Here is a sample for the DIII-D simulation (see also Tables (3), (4)):

```
160       NCYCLE   Number of GK calls; Number of Time Steps NTS=NCYCLE*NSUBCY
  3       NSUBCY   Number of MHD calls per each GK call
 20       NOUT     Number of time sequences; total time steps=NTS*NOUT
 92       LR1      Maximum value should be LR1=4*LM
1.0D-5    ETA      standard value: 1.0d-5
1.0D-8    VISC     standard value: 1.0d-8
0.02      DT       standard value: 0.02
 30       NPRI     NTS/NPRI<=MAXPRI
1.000001  RWALL    resistive wall normalized radius
                   (parameter required but not used by fortran, give any real number)
1.0D10    TAUWAL   resistive wall characteristic time
                   (parameter required but not used by fortran, give any real number)
0.0D00    VEDGE    plasma bulk velocity at the edge
                   (parameter required but not used by fortran, give any real number)
0.0D00    CURAMP   current ramp (now ignored)
.TRUE.    FREZ00   l=1 (m,n)=(0,0)
.TRUE.    FREZ10   l=2 (m,n)=(1,0)
.FALSE.   EQUIL
.FALSE.   DROP
1         NPROFI  0 DEN=1; 1 DEN=DEN(RHOA,ALFA,BETA); 2 DEN=(Q/Q0)**(-2)
3.9173d0  ALFA
0.69776d0 BETA
0.6471d0  RHOA
0.7       AGROWTH ad hoc growing factor parameter
0.01      BGROWTH ad hoc growing factor parameter
0.05      CGROWTH ad hoc growing factor parameter
0.95      DGROWTH ad hoc growing factor parameter
1         ITAERSP 1 drives TAE, 2 drives RSP (requires GROWTH .ne. 0)
1.D0      SMOFAC  amplitude of the smoothing factor
(1.D-07,1.D-07) AMP complex amplitude factor for the initial perturbation
0.00D-0   GROWTH  amplitude of the ad hoc growing term
.TRUE.    GYRO    call gyrokinetic module
.FALSE.   CYLIN   .true. MHD cylindrical limit
.FALSE.   BISEC   .true. bisection allowed
0.D-2     SKIN    (el. skin depth; skin=0.D0 ==> el. inertia neglected)
0.10d0    epsil1  parameter used to reduce toroidal corrections at the edge
0.95d0    cgeps   parameter used to reduce toroidal corrections at the edge
0.025d0   cweps   parameter used to reduce toroidal corrections at the edge

$DIAPOS
          NRCHNL=6,
          RCHNL(1)=0.200,
          RCHNL(2)=0.300,
          RCHNL(3)=0.400,
          RCHNL(4)=0.500,
          RCHNL(5)=0.650,
          RCHNL(6)=0.800,
&END
```

Quantities	Comments
NCYCLE	number of calls of the gyrokinetic module for each of the NOUT time sequences
NSUBCY	Number of MHD calls per each gyrokinetic call; NCYCLE*NSUBCY is the number of calls of the MHD module for each of the NOUT time sequences
NOUT	number of time sequences; total time steps=NCYCLE*NSUBCY*NOUT
LR1	maximum number of modes per MHD fields which are read in the file INCOND: maximum value should be LR1=4*LM
ETA	normalized resistivity, i.e. the inverse of the Lundquist number S (the ratio between resistive and Alfvén times $S \equiv \tau_\eta/\tau_{A0}$, with $\tau_\eta = \mu_0 a^2/\eta$ and $\tau_{A0} \equiv \omega_{A0}^{-1}$)
VISC	similar to ETA parameter, but representing viscosity
DT	elementary time step
NPRI	some outputs are performed every NPRI time steps; NPRI must satisfy NTS/NPRI<=MAXPRI
RWALL	resistive wall normalized radius (parameter required but not used by fortran, give any real number)
TAUWAL	resistive wall characteristic time (parameter required but not used by fortran, give any real number)
VEDGE	plasma bulk velocity at the edge (parameter required but not used by fortran, give any real number)
CURAMP	current ramp (only significant if “equilibrium” is evolved)
FREZ00	logical variable, if .true. does not evolve (“freeze”) l=1 mode (0,0)
FREZ10	logical variable, if .true. does not evolve (“freeze”) l=2 mode (1,0)
EQUIL	logical variable, if .true. the code is used to compute an “equilibrium” (useful for nonlinear MHD runs)
DROP	logical variable, if EQUIL=.true. and DROP=.true. kinetic energy is removed by dropping $\phi_{m,n}$ by a fixed factor (useful for preparing nonlinear MHD runs)

Table 3: Structure of the file PARAM.

Quantities	Comments
NPROFI	switch for assigning the bulk density radial profile $\hat{\rho}$: NPROFI=0 → $\hat{\rho} = 1$; NPROFI=1 → $\hat{\rho} = \hat{\rho}(\alpha, \beta, \rho_a)$; NPROFI=2 → $\hat{\rho} = [q(r)/q(0)]^{-2}$ (aligned toroidal gap)
ALFA, BETA, RHOA	$\hat{\rho}(\alpha, \beta, \rho_a) = (1 - \rho_a) * (1 - r^\alpha)^\beta + \rho_a$
AGROWTH, BGROWTH, CGROWTH, DGROWTH	parameters for an "ad hoc" growing term added in the vorticity equation (to simulate some particle drive): for CGROWTH $\leq$ r $\leq$ (CGROWTH+ DGROWTH): GROWTHR(I)=GROWTH*EXP(-((R(I)-AGROWTH)**2/BGROWTH)), else GROWTHR(I)=0 (now obsolete)
ITAERSP	ITAERSP=1 should drive a (1,1), (2,1) TAE using the "ad hoc" driving term, ITAERSP=2 should drive a (1,1), (2,1) RSP (Resistive Shear Periodic mode), requires GROWTH.ne.0.0 and $n = 1$, $m = 1, 2$ modes
SMOFAC	amplitude of a "smoothing" factor to control numerical instabilities in the center (r=0) (hyper-resistivity and hyper-viscosity terms)
AMP	complex amplitude factor for the initial perturbation
GROWTH	amplitude of the "ad hoc" growing factor
GYRO	logical variable, ifGYRO=.true. the gyrokinetic module is called
CYLIN	logical variable, ifCYLIN=.true. MHD module considers cylindrical limit while gyrokinetic module retains finite ϵ correction
BISEC	logical variable, ifBISEC=.true. allow time bisection in the MHD module (and, hence, in the gyrokinetic one)
SKIN	electron skin depth (if 0, electron inertia neglected, now obsolete, not used)
epsil1	parameter used to reduce ϵ correction at the edge (occasionally used to control edge numerical instabilities arising from MHD module, see Eq. (8))
cgeps	parameter used to reduce ϵ correction at the edge (see Eq. (8))
cweps	parameter used to reduce ϵ correction at the edge (see Eq. (8))
namelist DIAPOS	NRCHNL=6 diagnostic output channels, at the radii RCHNL(i=1,6), giving Real and Imaginary part of $\phi_{m,n}(r_{chnl}, t)$.

Table 4: Structure of the file PARAM (continued).

The general time stepping in HMGC is as follow: the MHD (normalized) time step is given by the parameter DT. Each NSUBCY MHD time steps, the gyrokinetic module is called. This loop is performed NCYCLE times. Thus a total umber of time steps of NTS=NCYCLE*NSUBCY is performed for each time sequence. This time sequence is repeated NOUT times. Thus, the total number of time steps of a simulation is $N_{\mathrm{time-steps}} =$ NTS*NOUT $=$ NCYCLE*NSUBCY*NOUT and the total (normalized) time simulated is $T_{\mathrm{total}} =$ DT*NCYCLE*NSUBCY*NOUT. Schematically, using a fortran-like schema, these nested loops are as follows:

...

```
time=0.
do i=1, NOUT
  do j=1,NCYCLE
    call GyroKinetic module
    do k=1,NSUBCY
      time=time+dt
      call MHD module
    enddo
  enddo
enddo
```

...

The time is normalized to the inverse of the on-axis ($r = 0$) Alfvén frequency, that is $t_{\mathtt{code}} = t(\mathrm{s})\omega_{A0}(\mathrm{s}^{-1})$ with $\omega_{A0} \equiv v_{A0}/R_0$ and v_{A0} the on-axis Alfvén velocity. If FREZ00=.true. and FREZ10=.true. (no evolution of equilibrium components) and a single perturbed n is included in the simulation, a linear MHD simulation will be performed: this is the usual operation condition of the MHD module. The parameters epsil1, cgeps and cweps are used to define a radial function $f_\epsilon(r)$ which modulates the toroidal corrections:

$$\begin{aligned} f_\epsilon(r) &= 1 - \left(\frac{1-\epsilon_1/\epsilon}{2}\right)\left[\tanh\left(\frac{r-cg_\epsilon}{cw_\epsilon}\right)+1\right] , \\ f_\epsilon(r) &\longrightarrow 1 \qquad \text{for} \qquad r \ll cg_\epsilon \\ f_\epsilon(r) &\longrightarrow \epsilon_1/\epsilon \qquad \text{for} \qquad r \gg cg_\epsilon \, . \end{aligned} \tag{8}$$

3.3 Input file stop_run

At the beginning, this file contains only one line, which can be used to overwrite the parameter NOUT given in the file PARAM:

```
20 nout_new
```

At run time, the same file is written and read by HMGC and this allows to stop or extend the run by editing it and changing the parameter nout:

```
  25  nout
  21  ncount
 192.0000000  time
```

Here ncount is the actual number of time sequences performed by the code and time is the corresponding normalized simulation time.

3.4 Input file TMODE

This input file contains the list of the Fourier modes actually included in the simulation. The first column is the mode index l, running from $l = 1$ to $l =$ `LM`. In the second and third columns are listed the corresponding poloidal ($m = m(l)$) and toroidal ($n = n(l)$) mode numbers. It is assumed that the modes are ordered by increasing n, and for each n by increasing m.

...

```
 1  0  0
 2  1  0
 3  1  2
 4  2  2
 5  3  2
 6  4  2
 7  5  2
 8  6  2
 9  7  2
10  8  2
11  9  2
12 10  2
13 11  2
14 12  2
15 13  2
16 14  2
17 15  2
18 16  2
19 17  2
20 18  2
21 19  2
22 20  2
23 21  2
```

...

3.5 Input file TMODE_PL

This input file is used by the plot post-processing program epe3ak31.u. It is exactly the same than the input file TMODE, with the add of a fourth column, in which 0 means that this component will not be considered by the plot program, whereas 1 means that it will be considered.

```
 1  0  0  0
 2  1  0  0
 3  1  2  1
 4  2  2  1
 5  3  2  1
 6  4  2  1
 7  5  2  1
 8  6  2  1
 9  7  2  1
10  8  2  1
11  9  2  1
12 10  2  1
13 11  2  1
14 12  2  1
15 13  2  1
16 14  2  1
17 15  2  1
18 16  2  1
19 17  2  1
20 18  2  1
21 19  2  1
22 20  2  1
23 21  2  1
```

3.6 Include file grid_inc

This lines of fortran are included in the MHD module (e3_complete.F) and in the gyrokinetic module (trough the general common commr31_complete.F) of HMGC. It defines general particle simulation parameters (see also Table (5)). Note that some of the following parameters refer to toroidal and poloidal meshes, also if the code is run in the nogrid mode (see Ref. [6]). Particles here are “simulation particles”. The file grid_inc is constructed from the two files nlr_inc and altri_grid_inc_1 written in the execution script:

...

```
cat >${HOMEroot_sources}/${pwr}_version/nlr_inc <<'EOF'
      PARAMETER(nlr=64)
EOF
```

...

...

```
cat >${HOMEroot_sources}/${pwr}_version/altri_grid_inc_1 <<'EOF'
      PARAMETER(NTH=168,
     &          nintphi=2*(nmax-1),
     &          nph_su_nintphi=4,
     &          NPH=nph_su_nintphi*nintphi,
     &          ippc=2,
     &          nne=672,
     &          npart=nlr*nth*nph*ippc**3,
     &          NMODOM=27,
     &          NRZ=5)
EOF
```

...

Please note that NLR should be NLR $\leq$ NR (see Sect. (3.1) where NR is defined). NTH should be chosen such that NTH $> 2m_{\max}$ (see Sect. (3)): in the following example, NTH $= 8m_{\max} = 8 * 21 = 168$ has been used. The factor 2 in the variable nintphi (nintphi=2*(nmax-1)) and the value of 4 for the variable nph_su_nintphi are such that NPH is NPH $= 8n_{\max} = 8 * 2 = 16$. Those values are the ones typically used in HMGC simulations.

The quantity nne should be such that npart = nne*nnalpha = nlr*nth*nph*ippc**3: a simple program to compute the optimal values to distribute the particles in the (E, α) space is provided (calcolo_nne.f), for given nlr, nth, maximum n, nph_su_nintphi, ippc. The program asks at the beginning which source you are referring to: enter “0” for data referring to e3_complete.

Quantities	Comments
`NLR`	number of radial cells for the gyrokinetic module (`NLR+1` mesh points)
`NTH`	number of points in the θ (poloidal angle) direction
`nintphi`	`nintphi=2*(nmax-1).` Parameter for φ (toroidal angle) mesh: the energetic particle distribution function will be loaded on `nph_su_nintphi` toroidal mesh points and then replicated on `nintphi-1` remaining sectors
`nph_su_nintphi`	parameter for φ (toroidal angle) mesh
`NPH`	`NPH=nph_su_nintphi*nintphi`: number of points in the φ (toroidal angle) direction
`ippc`	number of particles per cell in each direction of the physical space (r, θ, φ)
`nne`	number of particles in the energy space (`nne*nnalpha = npart/nintphi`, `nnalpha` is the number of particles in the pitch angle direction)
`npart`	`nlr*nth*nph*ippc**3`: total number of particles
`NMODOM`	number of Fourier components for the gyrokinetic module: they must be the Fourier modes of the MHD module plus two poloidal satellite modes for each toroidal mode considered in the simulation
`NRZ`	the particles are evolved in a (`NRZ,NRZ`) grid in the (R, Z) plane around the $r = 0$ point to avoid problems related to the singular point $r = 0$ in polar coordinates.

Table 5: Structure of the file `grid_inc`.

3.7 Input file KININP

This file is the main parameter input for the gyrokinetic module. Here is a sample for the DIII-D simulation (see also Tables (6), (7)):

...

```
 0.032863457d0 RHOSA           idistr=1: sqrt(T_H0/m_H)/Omega_cH0/a;
                               idistr=2: sqrt(E_0/m_H)/Omega_cH0/a;
                               idistr=3: sqrt(T_perp_H0/m_H)/Omega_cH0/a
 0.271063836d0 VTHSVA          idistr=1: sqrt(T_H0/m_H)/v_A0;
                               idistr=2: sqrt(E_0/m_H)/v_A0);
                               idistr=3: sqrt(T_perp_H0/m_H)/v_A0
 1.5d0         sigma_0         idistr=3: T_perp_H0/T_par_H0
 2.3256d0      usdelta_input   idistr=1,2: anisotropy parameter (1/width)
 0.68128d0     cosalfa_0_input idistr=1,2: anisotropy parameter
                               (cosine of injection pitch angle)
 4.153850158d0 e0sec0          idistr=2: on-axis E0/E_crit0
 .2D00         ALF             parameter controlling non uniform radial
                               particle loading
  0            NPIC            parameter controlling non uniform radial
                               particle loading
                               (if npic.ne.0, er0(i), del(i), i=1,npic must follow)
 0.264848976d0 ENHSNI          n_H0/n_i0 on axis ratio between energetic particle
                               and bulk ion densities
 1.D00         EMHSMI          m_H/m_i   ratio between energetic particle and bulk
                               ion masses
 999.99d9      timkin_anu      energetic particle density ramps for t > timkin_anu
                               (set to a large value to avoid ramping)
 120.d0        TIMKIN_RELAX    time at which the distribution function is assumed
                               to be relaxed; used for ramping and diagnostics
 1.166212d0    ANU_MAX         energetic particle density ramping parameter
 0.166212d-2   ANU_DOT_0       energetic particle density ramping parameter
 0             i_write_deltaf  0: no output,
                               1: output of f(r,mu,u,t),
                               2: output of f(r,E,alpha,t)
 0             i_write_power   0: no output,
                               1: output of wave-particle power exchange P(r,mu,u,t)
 30            NWRITE          energetic particle quantities written every
                               NWRITE time steps
 2             IDISTR          1: Maxwellian, 2: Slowing-down, 3: bi-Maxwellian
 1             IDELTF          0: full f, 1: delta-f
 0             ILIN            0: fully non linear GK simulation,
                               1: linear GK simulation
 0             IRLSR0          0: R_l with epsilon corrections, 1: R_l=R0
 1             IMIRR           0: mirroring term off, 1: mirroring term on
 1             IW00            0: grad-B drift contribution to the source term
                               neglected, 1: retained
 1             ILANDA          0: Landau damping term off, 1: Landau damping term on
 1             ICURV           0: curvature term off, 1: curvature term on
 1             IOMST           0: omega-star term off, 1: omega-star term on
 2             NPTEST          number of test particles
 0             ITEST           0: true test particle, 1: simulation particle
.5,0.,0.,.1,1. EROT,THOT,PHOT,AMOT,UOT for t=0: r,theta,phi,mu,u
 1             ITEST           0: true test particle, 1: simulation particle
1352           LTEST           simulation particle identification number
0.             ER_PERT0:       energetic particle pressure term PREK set to
                               zero for r<er_pert0
1 1            l, iprek(l)     iprek(l)=0: l-th Fourier component of the PREK off,
                               iprek(l)=1: on
2 1            l, iprek(l)
```

...

Quantities	Comments
RHOSA	ρ_H/a: particle Larmor radius normalized to minor radius computed: (1) at the on-axis energetic particle temperature T_{H0} ($\rho_H/a = (\sqrt{T_{H0}/m_H}/\Omega_{cH0})/a$ if a Maxwellian distribution is considered, IDISTR=1); (2) at the birth energy E_0 ($\rho_H/a = (\sqrt{E_0/m_H}/\Omega_{cH0})/a$) if a slowing down distribution function is assumed, IDISTR=2); (3) at the on-axis perpendicular energetic particle temperature $T_{\perp,H0}$ ($\rho_H/a = (\sqrt{T_{\perp,H0}/m_H}/\Omega_{cH0})/a$ if a bi-Maxwellian distribution is considered, IDISTR=3)

VTHSVA	v_{th}/v_{A0}: ratio between energetic particle thermal velocity and on-axis Alfvén velocity:
	for IDISTR=1: $v_{th}/v_{A0} = \sqrt{T_{H0}/m_H}/v_{A0}$;
	for IDISTR=2: $v_{th}/v_{A0} = \sqrt{E_0/m_H}/v_{A0}$;
	for IDISTR=3: $v_{th}/v_{A0} = \sqrt{T_{\perp,H0}/m_H}/v_{A0}$

sigma_0	ratio between on-axis perpendicular and parallel energetic particle temperatures (used only if IDISTR=3)
usdelta_input	parameter for anisotropic particle distribution function, it corresponds to the inverse of the width Δ (see Sects. (6.1), (6.2)) of the distribution function around the injection pitch angle α_0 for Maxwellian or slowing down distribution functions (IDISTR=1 or IDISTR=2)
cosalfa_0_input	$\cos\alpha_0$, cosine of the injection pitch angle α_0 for Maxwellian or slowing down distribution functions (IDISTR=1 or IDISTR=2)
e0sec0	$E_0/E_{crit,0}$, on-axis ratio between birth energy and critical energy for slowing down distribution function (IDISTR=2)
ALF	parameter for non uniform energetic particles radial loading: uniform fraction, 0<ALF<1
NPIC ER0(I),DEL(I)	parameter for non uniform energetic particles radial loading: number of gaussians overimposed to the uniform distribution fraction ALF. If NPIC.ne.0 the corresponding values of the radial positions and widths (ER0(I),DEL(I)) of the gaussians must be given

Table 6: Structure of the file KININP.

Quantities	Comments
ENHSNI	n_{H0}/n_{i0}, on-axis ratio between the energetic particle and bulk ion densities
EMHSMI	m_H/m_i, ratio between the energetic particle and bulk ion mass
timkin_anu	parameter for ramping the energetic particle density: energetic particle density ramps for $t >$ timkin_anu. timkin_anu should be greater than TIMKIN_RELAX but smaller than the time at which non linear phase occurs. Set timkin_anu greater than the total simulation time to avoid ramping
TIMKIN_RELAX	time at which the energetic particle distribution function is assumed to be relaxed (because of initialization of the distribution function in terms of non conserved quantities)
ANU_MAX	parameter for ramping the energetic particles: multiplying factor of the normalized energetic particle density
ANU_DOT_0	parameter for ramping the energetic particles: time derivative of the normalized energetic particle density
i_write_deltaf	produces the output of the distribution function: i_write_deltaf=0: no output is produced, i_write_deltaf=1: $f(r, \mu, u, t)$, i_write_deltaf=2: $f(r, E, \alpha, t)$ (μ, u, E and α are the magnetic moment, parallel velocity, particle energy and pitch angle, respectively)
i_write_power	flag for writing the power exchange $P(r, \mu, u, t)$ between particles and waves: 0: no output is produced, 1: output is produced
NWRITE	energetic particle quantities are written on output files every NWRITE*NSUBCY time steps
IDISTR	1: Maxwellian, 2: Slowing-down, 3: bi-Maxwellian
IDELTF	0: performs a full-f simulation, 1: performs a δf simulation (standard use of HMGC)

Table 7: Structure of the file KININP (continued).

Quantities	Comments
ILIN	0: performs a gyrokinetic non-linear simulation (standard use of HMGC), 1: performs a gyrokinetic linear simulation (note that ILIN=1 corresponds exactly to a linear simulation only if the initial distribution function is a true equilibrium function)
IRLSRO	for the generic l simulation particle: 0: retains ϵ corrections $R_l = R_0(1+\epsilon_l \cos\theta_l)$ (standard use of HMGC), 1: approximates $R_l = R_0$
IMIRR	0: mirroring terms off, 1: mirroring terms on (standard use of HMGC)
IW00	IW00=0 causes to neglect the contribution of the grad-B drift to the source term in the delta-f Vlasov equation 1: terms are retained (standard use of HMGC)
ILANDA	0: Landau damping off, 1: Landau damping on (standard use of HMGC)
ICURV	0: curvature term off, 1: curvature term on (standard use of HMGC)
IOMST	0: ω_* term off, 1: ω_* term on (standard use of HMGC)
NPTEST	number of test particles
ITEST	for each test particle enter: 0: to initialize a true test particle, 1: to follow a simulation particle
5 reals, *or* LTEST	if ITEST=0, the initial coordinates ($t = 0$) of the test particle must be given ($r_{\mathrm{test}}, \theta_{\mathrm{test}}, \varphi_{\mathrm{test}}, \mu_{\mathrm{test}}, u_{\mathrm{test}}$), if ITEST=1, particle identification number
ER_PERTO	energetic particle pressure tensor term (PREK) set to zero for $r \leq$ ER_PERTO (standard use of HMGC: ER_PERTO=0)
l, iprek(l)	PREK(l)=0: l^{th} Fourier component inactive, PREK(l)=1: l^{th} Fourier component of the energetic particle pressure tensor term active (defaults is all Fourier components active)

Table 8: Structure of the file KININP (continued, 2).

3.8 Input files: energetic particle density and temperature profiles

Normalized (to the on-axis value) energetic particle density profile and temperature (if Maxwellian distribution function is loaded, `idistr=1`), E_{crit} (if slowing down distribution function is loaded, `idistr=2`) or perpendicular and parallel temperatures (if bi-Maxwellian distribution function is loaded, `idistr=3`) profiles must be provided on an equally spaced normalized poloidal flux function grid ψ. These profiles are usually provided by standard transport codes (e.g., available in the ITER database). If experimental profiles are provided (e.g., $\rho, q(\rho), n_H(\rho), T_e(\rho), T_H(\rho)$), as functions of $\rho \equiv \sqrt{\Phi/\Phi_{\text{limiter}}}$, the usual radial-like coordinate of transport codes (e.g., TRANSP) with Φ the toroidal magnetic flux function, a simple program is provided (`psi_from_rho_q_exp.f`) which returns the poloidal flux function ψ in terms of ρ, integrating the following expression:

$$2\pi d\psi = \frac{d\Phi}{q(\rho)}\,, \tag{9}$$

to obtain $\psi = \psi(\rho)$. The normalized coordinate proportional to the poloidal flux function should be such that it is zero in the centre and unity at the edge.

Those profiles will be interpolated using splines on the desired equally spaced normalized ψ mesh by the fortran program `interp_spline.f`. “Experimental” files with $\psi_{\text{norm}}(\rho)$, $n_{H,\text{norm}}(\rho)$ (and similar for the other quantities) must be provided (their names are, e.g., `den_exp_DIII_D_1`, `temp_exp_DIII_D_1`, `temp_exp_par_xxx`, where `temp_exp_DIII_D_1` contains the energetic particle isotropic temperature in the case `idistr=1`, the E_{crit} normalized profile in the case `idistr=2`, or the energetic particle perpendicular (`temp_exp_DIII_D_1`) and parallel (`temp_exp_par_xxx`) temperature profiles in the case `idistr=3`). Then a corresponding output on the equally spaced ψ_{norm} mesh will be produced.

4 Output files: produced by the MHD module

4.1 Output file CTTO

The CTTO file has exactly the same form of the INCOND file, but includes all the Fourier components.

4.2 Output file ENERGY

The ENERGY file contains the four namelists params, diapos, equipa, paramk. Then it contains a time sequence of some global fluid quantities, namely:

```
      WRITE(CHENER,00003)
     &   TIMBUF(JTBUF),LM,EZBUF(JTBUF),QOBUF(JTBUF),
     &   QABUF(JTBUF),2.*QABUF(JTBUF)**2*ENMODE(JTBUF,1,1),
     &   (ENBUF(JTBUF,I),I=1,4),
     &   (RM(L),RN(L),(ENMODE(JTBUF,K,L),K=1,IDIAGN+2),L=1,LM)
00003 FORMAT(F16.6,I6,E16.6,3F10.6,/,
     &       4E24.15,/,(2F5.0,4E24.15,/,10X,4E24.15,/,
     &                    10X,4E24.15,/,10X,2E24.15))
```

where the meaning of the quantities are listed in Table (9). The MHD module produces an output on the file ENERGY every NPRI time steps, thus the output time step is given by $\Delta t_{\mathrm{output,MHD}}$ = DT*NPRI (in Alfvén time units).

Quantities	Comments
`TIMBUF`	simulation time t
`LM`	Fourier components considered in the simulation
`EZBUF`	electric field (toroidal corrections neglected) at r=1 (significative only for nonlinear MHD simulations)
`Q0BUF`	$q(r = 0)$
`QABUF`	$q(r = 1)$
`2.*QABUF...`	internal inductance l_i (toroidal corrections neglected)
`ENBUF(...,I)`	`I=1`: volume integrated total magnetic energy `I=2`: volume integrated total kinetic energy `I=3`: resistive dissipation (obsolete, not corrected for toroidal terms) `I=4`: viscous dissipation (obsolete, not corrected for toroidal terms)
`RM(L)`	$m(l)$
`RN(L)`	$n(l)$
`ENMODE(...,K,L)`	`K=1`: volume integrated magnetic energy of the l^{th} Fourier component, `K=2`: volume integrated kinetic energy of the l^{th} Fourier component, `K=3,IDIAGN+2`: real and imaginary part of the $\phi_{m(l),n(l)}$ at specific diagnostic radii, as given in namelist `diapos` (now `IDIAGN=12=2*NRCHNL`, see Sect. (3.2).)

Table 9: Quantities in the file `ENERGY`.

This file is read by the plotting program `epe3ak31.u` and give global time evolution of the simulation from the point of view of the MHD module (see Fig. (4)). An example of the input file `xepe3ak31_datain` for the program `epe3ak31.u` is listed hereafter:

———————————— ... ————————————

```
ENERGY
0              CHANGE TSTART (IF 1 ADD IN THE NEXT LINE NEW TSTART)
0              CHANGE TEND   (IF 1 ADD IN THE NEXT LINE NEW TEND)

1              MODECH (1-MODES , 2-ENERGY , 3-PARAMETERS 4-MAGNETIC SIGNALS)
1              IC (1-MAGNETIC , 2-KINETIC , 3-TOTAL)
0              CHANGE MODE LIST (IF 1 ENTER SEQUENCE OF MODE CHANGE)
1              CHANGE LIMITS (IF 1 NEW LIMITS ARE ENTERED BY TERMINAL)
1.e-30,1.e-4

1              MODECH (1-MODES , 2-ENERGY , 3-PARAMETERS 4-MAGNETIC SIGNALS)
2              IC (1-MAGNETIC , 2-KINETIC , 3-TOTAL)
0              CHANGE MODE LIST (IF 1 ENTER SEQUENCE OF MODE CHANGE)
1              CHANGE LIMITS (IF 1 NEW LIMITS ARE ENTERED BY TERMINAL)
1.e-30,1.e-4

1              MODECH (1-MODES , 2-ENERGY , 3-PARAMETERS 4-MAGNETIC SIGNALS)
3              IC (1-MAGNETIC , 2-KINETIC , 3-TOTAL)
0              CHANGE MODE LIST (IF 1 ENTER SEQUENCE OF MODE CHANGE)
1              CHANGE LIMITS (IF 1 NEW LIMITS ARE ENTERED BY TERMINAL)
1.e-30,1.e-4

0              exit
```

———————————— ... ————————————

Note that in the above list only the sequence of input data suited for `MODECH=1` have been shown; different values of `MODECH` will provide plots of other quantities and will require different input data.

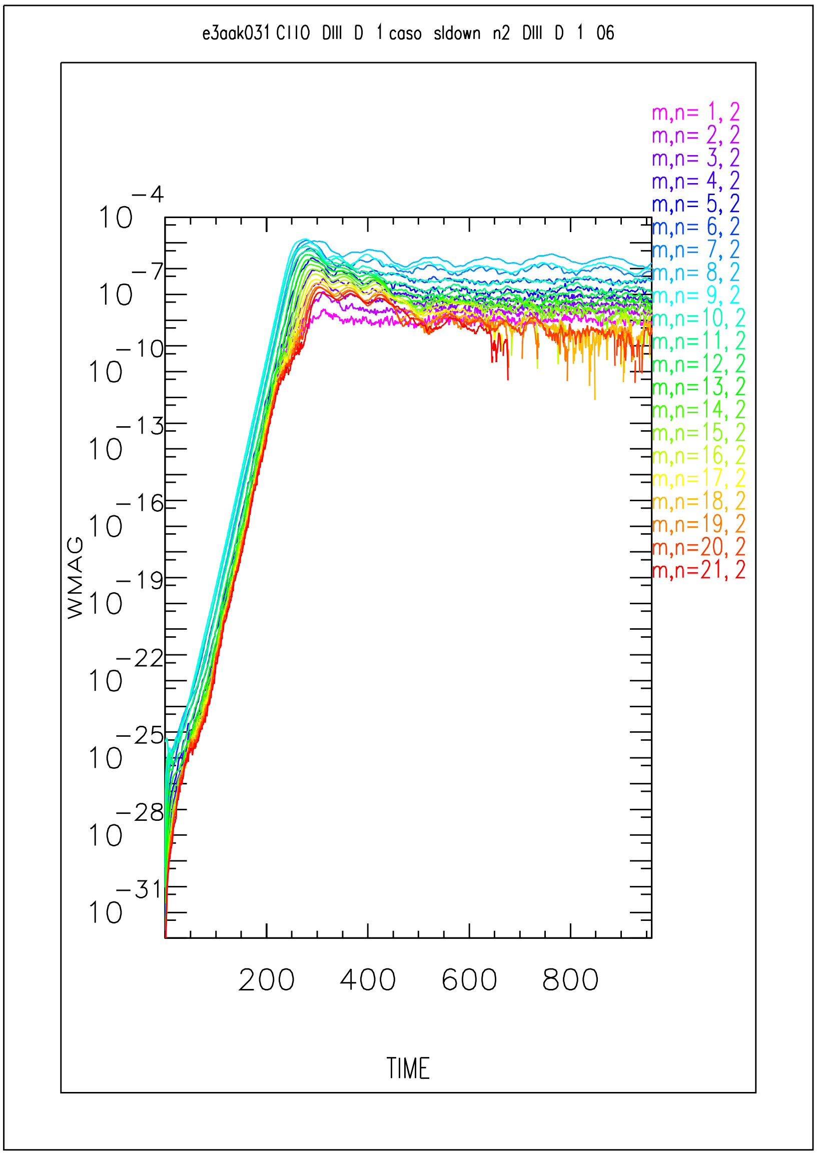

Figure 4: Radially integrated magnetic energy of the perturbed Fourier components vs. time: the flag (fourth column) in the `TMODE_PL` input file commands the Fourier components to be plotted. Note that here the result of a simulation in which `nout`=100. is shown.

After running the program epe3ak31.u, an output file is written which can be used to produce a time sequence of plots (movie) by the fortran plot_energy.f. An example of the input file xplot_energy_input for the program plot_energy.f is listed hereafter:

——————————— ... ———————————

```
ENERGY
TMODE_PL
 0          ips 0: no PostScript file, 1: PS file, 2: EPS, name follows (30 char.)
pippo.eps
 1, 321, 10 ifirst_step,itot,increm (output time steps)
 3          1=magnetic energy, 2=kinetic energy, 3=magnetic + kinetic energy
    0.      time_max (if 0, take time_max from stop_run file)
 1.e-32, 0. y_en_min, y_en_max
  161       ipl0 first plot
  161       ipl1 last  plot
 5
```

——————————— ... ———————————

The above input data example contains: the name of the input data files (ENERGY, TMODE_PL). ips is a flag for producing a PostScript output, followed by the name of such a file (pippo.eps). ifirst_step,itot are the first and last output time steps which will be read and available for plotting, respectively; frames will be produced every increm steps. An output time step corresponds to NPRI simulation time steps, i.e. to a simulation time $\Delta t_{\mathrm{output,MHD}}$ = DT*NPRI (in Alfvén time units). time_max is the limit of the abscissa (in Alfvén time units, if time_max=0., the maximum of the abscissa will be calculated automatically by the input parameters of the run). Note that for a simulation with DT=0.02, NCYCLE=160, NSUBCY=3, NPRI=30, NOUT=20 we get a total number of time steps equal to NCYCLE*NSUBCY*NOUT+1=9601 (t=0 is also counted) corresponding to time_max=DT*NCYCLE*NSUBCY*NOUT=192.0, and (NCYCLE*NSUBCY*NOUT/NPRI)+1=321 output times. y_en_min, y_en_max are the limits of the ordinate (if y_en_min=0. and/or y_en_max=0., these limits are computed from the data). ipl0 and ipl1 are the first and last time steps at which graphical frames will be produced (in the above example, only the frame number "161" will be produced, corresponding to the normalized time $t\omega_{A0} = (161-1)$*DT*NPRI $= 96.$). The parameter "5" is a parameter required by the plotting routines (HIGZ from CERN) which identify the graphic window.

In Fig. (5) an example of this plot is shown:

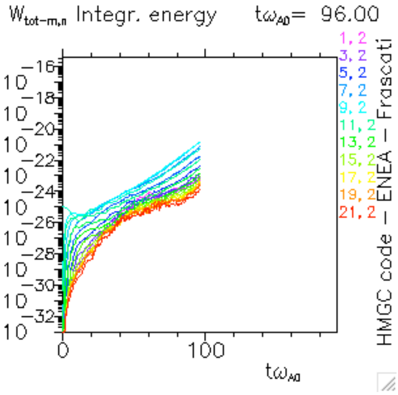

Figure 5: Example of integrated total energy $W_{\mathrm{tot}-m,n}$ vs. time. The labels on the right of the plot, describing the (m, n) poloidal and toroidal mode numbers of the curves, are written every two modes, because of space limitation (although in the plot all the components are shown).

4.3 Output file `CTBO`

The `CTBO` file contains radial profile data of MHD quantities, at the end of the simulation (or each `NCYCLE*NSUBCY` times, if fortran is modified accordingly).

...

```
      WRITE(CHCTBO,1900) ETA,VISC,DT,TIME,REAL(NR),REAL(LM),REAL(IRMODE)
      WRITE(CHCTBO,1900) (RM(L),RN(L),L=1,LM)
      WRITE(CHCTBO,1900) (R(I),I=1,NR)
 1900 FORMAT(6(E20.13,','))
c...
      WRITE(CHCTBO,1950) ((PSI(I,L),I=1,NP),L=1,LM)
      WRITE(CHCTBO,1950) ((PHI(I,L),I=1,NR),L=1,LM)
CGV
      WRITE(CHCTBO,1950) ((CUR(I,L),I=1,NR),L=1,LM)
      WRITE(CHCTBO,1950) ((  W(I,L),I=1,NR),L=1,LM)
CGVKIN..
      WRITE(CHCTBO,1950) ((PREK(I,L),I=1,NR),L=1,LM)
CGVKIN..
CGV
 1950 FORMAT(3('(',E20.13,',',E20.13,'),'))
```

...

The variable IRMODE is an obsolete parameter, included only for compatibility with old outputs. This output file is read by the plotting program `profilk.u` and gives the radial profile for each Fourier poloidal component of the various MHD variables (the poloidal magnetic flux function `PSI` $\equiv \psi_{m,n}(r)$, the scalar potential `PHI` $\equiv \phi_{m,n}(r)$, the toroidal component of the current `CUR` $\equiv j_{m,n}(r) \equiv -(\triangle^* \psi)_{m,n}$, the toroidal component of the vorticity `W` $\equiv w_{m,n}(r) \equiv (\nabla^2_\perp \phi)_{m,n}$, and term proportional to the divergence of the energetic particle stress tensor which enters in the vorticity equation `PREK`$_{m,n}(r)$, computed at the time t=`TIME`, see Fig. (6)). An example of the input file `xprofilk_datain` for the program `profilk.u` is listed hereafter:

...

```
0
0
title
CTBO

1              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
1              L (mode index) (if l<0, it plots from abs(l) to lm;
                                 do not enter min,max)
0.,1.,,,       X-MIN,X-MAX,Y-MIN,Y-MAX (if commas, it takes computed values)

1              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
2              L(M,N) (if l<0, it plots from abs(l) to lm; do not enter min,max)
0.,1.,,,       X-MIN,X-MAX,Y-MIN,Y-MAX (if commas, it takes computed values)

8              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT

1              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
-3             L(M,N) (if l<0, it plots from abs(l) to lm; do not enter min,max)

8              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
```

```
3              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
-3             L(M,N) (if l<0, it plots from abs(l) to lm; do not enter min,max)

8              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT

6              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
-3             L(M,N) (if l<0, it plots from abs(l) to lm; do not enter min,max)

8              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT

2              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
-1             L(M,N) (if l<0, it plots from abs(l) to lm; do not enter min,max)

8              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT

4              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
-1             L(M,N) (if l<0, it plots from abs(l) to lm; do not enter min,max)

7              1-PSI 2-W   3-PHI 4-J 5-Q 6-PREK 7-NEW TIME 8-BLANK PLOT
```

...

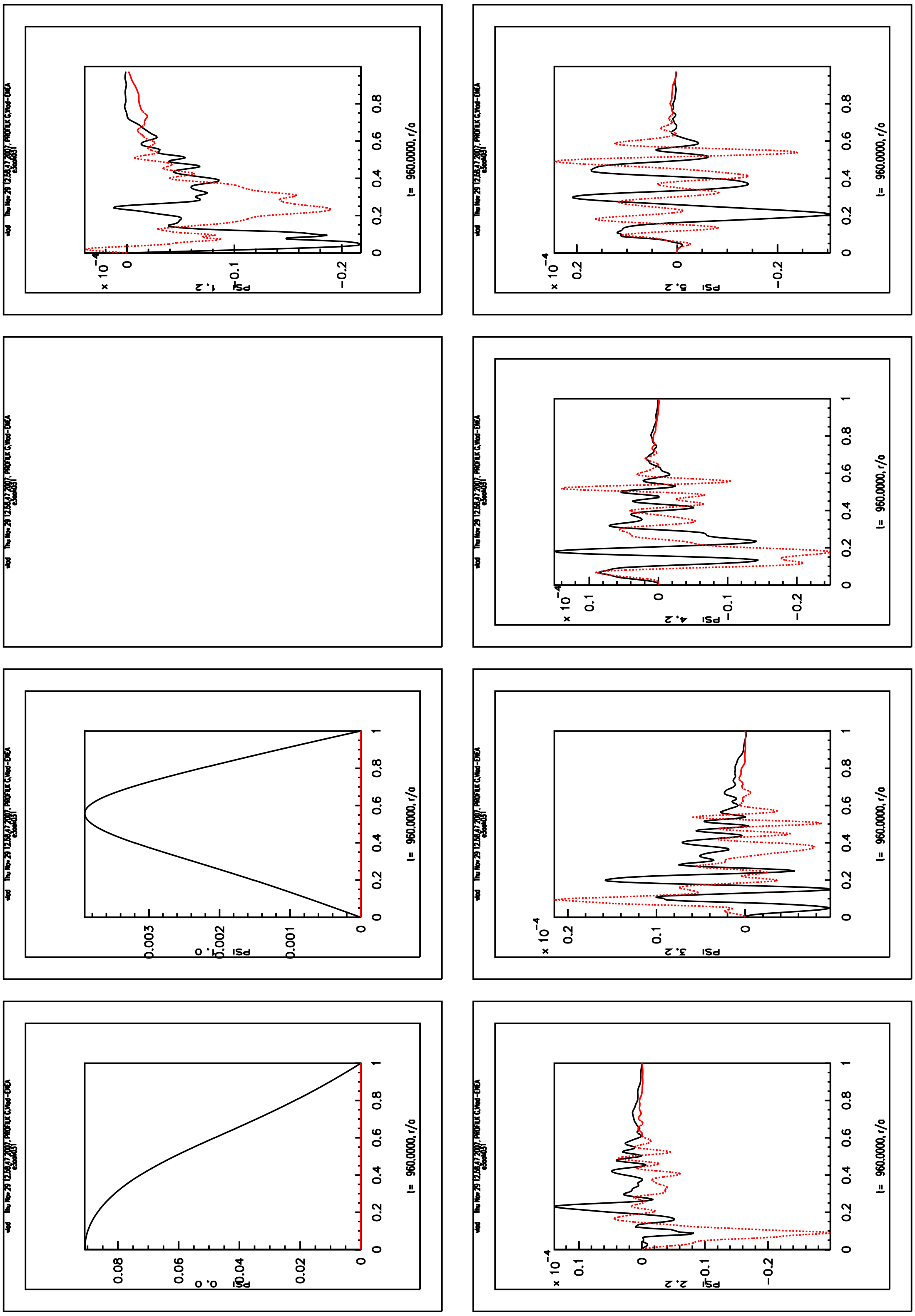

Figure 6: Radial profiles of the Fourier components from `CTBO` file as produced by the previous script (only the first 8 frames are shown): solid black line: real part, dotted red line: imaginary part.

5 Output files: produced by the gyrokinetic module

5.1 Output file OUTDNT

This file contains the time series of the radial profiles of energetic particle density DNTOT (normalized energetic particle density), perpendicular PPERP and parallel PPARA energetic particle pressure. The gyrokinetic module produces an output on the file OUTDNT every NWRITE time steps (in analogy with the MHD module, with NWRITE taking the place of NPRI), thus the output time step is given by $\Delta t_{\text{output,GK}} =$ DT*NWRITE (in Alfvén time units).

...

```
      WRITE(43,*)ISTEP0,TIMKIN,DENOUT
      IF(ISTEP0.EQ.0)THEN
        WRITE(43,*)ASPECT
        WRITE(43,*)USPS00
        write(43,*)' 0'
        WRITE(43,*)NERRE,LMEQ
        DO IR=0,NERRE
          WRITE(43,*)RLOW(IR)
        ENDDO
c
        DO IR=0,NERRE
          DO LMODE=1,LMEQ
            WRITE(43,*)PSIEQ(LMODE,IR)
        ENDDO
        ENDDO
c
        DO LMODE=1,LMEQ
          WRITE(43,*)MMODE(LMODE)
        ENDDO
      ENDIF
c...
      DO 1 JJER=0,NLR
      WRITE(43,*)DNTOT(JJER),ANTOT,PPERP(JJER),PPARA(JJER)
    1 CONTINUE
```

...

ISTEP0 is the current time step index, TIMKIN is the (normalized) current time, DENOUT=0.0, USPS00= $(\psi_{0,0}(r)|_{max})^{-1}$, ASPECT is the aspect ratio R_0/a, NERRE=NR-1, RLOW is the radial coordinate of the mesh used by the MHD module, PSIEQ are the radial profiles of the LMEQ equilibrium Fourier components of ψ, MMODE(l) $= m(l)$, ANTOT is an obsolete quantity. This file is used to produce movies of density, β_H and α_H (local drive) energetic particle profiles evolution (see Fig. (7)) using the plot program plot_density.f. An example of the input file xplot_density_input for the program plot_density.f is listed hereafter:

...

```
ENERGY
APWRITE
OUTDNT
 0          ips 0: no PostScript file, 1: PS file, 2: EPS, name follows (30 char.)
pippo.eps
 1, 321, 10 ifirst_step,itot,increm (output time steps)
 0          1=plot n_H-density, 2= beta_H, 3= alpha_H, 0= all
  161       ipl0 first plot
  161       ipl1 last  plot
 0.  0.     xldmin, xldmax (x axis, if 0.,0. use automatic values)
 0.  1.20   aldmin, aldmax (density, if 0.,0. use automatic values)
 0.  0.011  albmin, albmax (beta_H, if 0.,0. use automatic values)
-1.5 1.5    alamin, alamax (alpha_H, if 0.,0. use automatic values)
 5
```

...

In the above example, which refers to a simulation with DT=0.02, NCYCLE=160, NSUBCY=3, NWRITE=30, NOUT=20, we get a total number of time steps equal to NCYCLE*NSUBCY*NOUT+1=9601 (t=0 is also counted) corresponding to time_max=DT*NCYCLE*NSUBCY*NOUT=192.0, and (NCYCLE*NSUBCY*NOUT/NWRITE)+1=321 output times. Only the plots corresponding to the output time step ipl0 = ipl1 = 161 will be produced, that is at the normalized time $t\omega_{A0} = (161-1)$*DT*NWRITE $= 96$. The parameter "5" in the last line of the file xplot_density_input is a parameter required by the plotting routines (HIGZ from CERN) which identifies the graphic window.

The plotted quantities are given in terms of the quantities written in the file OUTDNT by:

- $n_H(r)/n_{H0} = \texttt{DNTOT}(r)/(2\pi\Delta r)$ where $\Delta r = 1/\texttt{NLR}$;
- $\beta_H(r) = 2 \times \texttt{EMHSMI} \times \texttt{VTHSVA}^2 \times \texttt{ENHSNI} \times [2 \times \texttt{PPERP}(r)/3 + \texttt{PPARA}(r)/3]/(2\pi\Delta r)$
- $\alpha_H = -R_0/a \times q^2(r) d\beta_H/dr$.

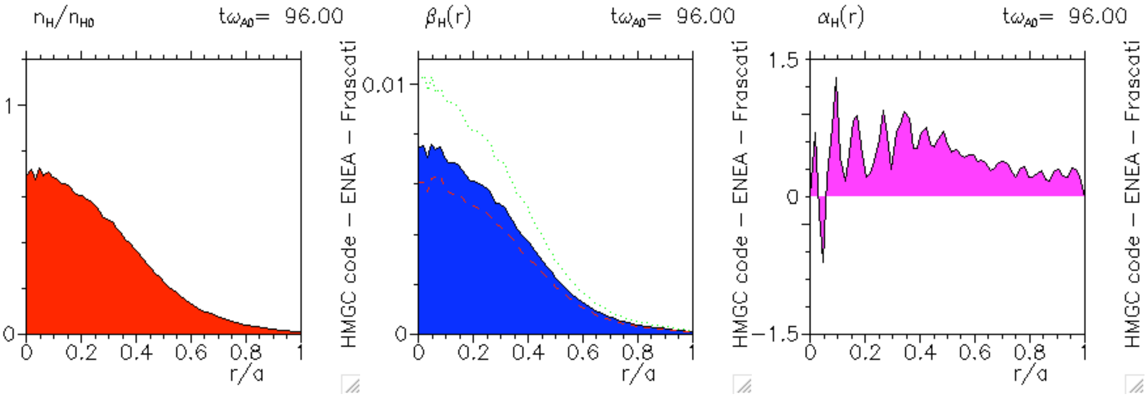

Figure 7: Example for density, β_H (red dashed curve is $\beta_{\perp,H}$, green dotted curve is $\beta_{\|,H}$) and α_H radial profiles.

5.2 Output file TESTWRIK

This file contains the time history of the test particles, and it is used to plot test particle quantities (see Fig. (8)) using the plot program `plot_field.f`. The quantities written are: $r_{\text{test}}, \theta_{\text{test}}, \varphi_{\text{test}}, u_{\text{test}}, w_{\text{test}}, \Delta_{\text{test}}$ (see Sect. (9.4)). The following excerpt shows the write statements in the file `TESTWRIK`:

...

```
      IF(ISTEP0.EQ.0)THEN
C-----------------------------------------------------------------------
      WRITE(29,*) ASPECT
      WRITE(29,*) NPTEST
      DO 335 I=1,NPTEST
      WRITE(29,*) ITEST(I)
      IF(ITEST(I).EQ.0)THEN
      WRITE(29,*) EROT(I),THOT(I),PHOT(I),AMOT(I),UOT(I)
      ELSE
      WRITE(29,*) LTEST(I)
      ENDIF
  335 CONTINUE
C-----------------------------------------------------------------------
      ENDIF
C-----------------------------------------------------------------------
C     WRITE THE TIME STEP AND THE QUANTITY DENOUT INTO FILE 29
C-----------------------------------------------------------------------
      WRITE(29,*)ISTEP0,TIMKIN,DENOUT
C-----------------------------------------------------------------------
C     WRITE THE RELEVANT DATA INTO FILE 29
C-----------------------------------------------------------------------
      DO 1 L=1,NPTEST
      WRITE(29,*)ERTEST(L),THTEST(L),PHTEST(L),UTEST(L)
      WRITE(29,*)WTEST(L),DTEST(L)
    1 CONTINUE
```

...

Units and normalizations are:

- `ERTEST` $\equiv r_{\text{test}}$ normalized to minor radius a;
- `THTEST` $\equiv \theta_{\text{test}}$ in radiants;
- `PHTEST` $\equiv \varphi_{\text{test}}$ in radiants;
- `UTEST` $\equiv u_{\text{test}} = v_{par}/v_H(\psi = 0)$, with $v_H(\psi = 0) = \sqrt{T_H(\psi = 0)/m_H}$. $T_H(\psi = 0)$ is:
  1. the temperature at $\psi = 0$ for Maxwellian distribution function (`idistr=1`);
  2. the birth energy E_0 for the slowing down distribution function (`idistr=2`);
  3. the perpendicular temperature at $\psi = 0$ for the bi-Maxwellian distribution function (`idistr=3`);
- `AMOT` $\equiv \mu * \Omega_H(\psi = 0)/T_H(\psi = 0)$, with μ being the magnetic moment and $\Omega_H(\psi = 0)$ the Larmor frequency of the energetic particle at $\psi = 0$;

Note that here $\psi = 0$ defines the magnetic axis.

5.3 Outputs file PHIWRITE and APWRITE

These files contain the time sequences of the radial profiles of the Fourier components of $\phi_{m,n}(r)$ and $\psi_{m,n}(r)$, respectively, on the radial grid of the gyrokinetic module and normalized according to the normalizations used therein. The files are in `unformatted` format. The following excerpts show the write statements in the file PHIWRITE (`unit=63`) and APWRITE (`unit=64`):

...

```
      WRITE(63)ISTEP0,TIMKIN
      IF(ISTEP0.EQ.0)THEN
        WRITE(63)NLR,NTH,NPH
        WRITE(63)(ERG(JER),JER=0,NLR)
        WRITE(63)LMPERT
        write(63) (mmode(im),nmode(im),im=1,lmpert)
      ENDIF
      write(63) ((phmhdg(im,jer),im=1,lmpert),jer=0,nlr)
      ...
```

...

...

```
      ...
      write(64) ((apmhdg(im,jer)+psieqg(im,jer),im=1,lmpert)
     &                ,jer=0,nlr)
      ...
```

...

Note that the file APWRITE contains the $\psi_{m,n}(r)$ components (perturbation part `apmhdg` plus equilibrium part `psieqg`) and the file PHIWRITE contains $\phi_{m,n}(r)$ (perturbation part `phmhdg`, equilibrium part assumed to be zero). ERG is radial coordinate of the mesh used by the the gyrokinetic module, LMPERT=LM. Those files are used to produce a sequence of frames of a series of quantities (only the ones produced for $\phi_{m,n}(r)$ are shown in the following Figs. (8), (10)), using the plotting program `plot_field.f`. The plots are the following: contour plot of $\phi(r, \theta, \varphi = \texttt{phiplot})$, $\phi(r, \theta, \varphi = \varphi_i(t))$ with superimposed the trajectory of the i^{th} test particle, trajectories of the i^{th} test particle $r_i = r_i(t)$, $\theta_i = \theta_i(t)$, $\varphi_i = \varphi_i(t)$, $u_i = u_i(t)$ (see Fig. (8)), trajectory of the i^{th} test particle in the poloidal (R, Z) and equatorial (X,Y) plane (see Fig. (9)), radial profiles of the $\phi_{m,n}$ Fourier components, contour plot of $\phi_{m,n}(r)$ in the plane (r, m) with superimposed the curve $m = nq(r)$, contour plot of the power spectrum $P(r, \omega)$ in the plane (r, ω) with superimposed the lower and upper Alfvén continua for the toroidal gap, $\xi_r(r)$ (radial component of the displacement) and $\delta T_e(r)$ (electron temperature fluctuation) assuming incompressible perturbations. The power spectrum $P(r, \omega)$ is defined as:

$$P(r,\omega) \;=\; \Sigma_{m,n} P_{m,n}(r,\omega) \propto \Sigma_{m,n} \left(|\phi_{m,n}(r,\omega)|^2 + |\phi_{m,n}(r,-\omega)|^2 \right) . \qquad (10)$$

An example of the input file `xplot_field_input` for the program `plot_field.f` is listed hereafter:

```
ENERGY
PHIWRITE
APWRITE
TESTWRIK
Te_vs_erg_DIII_D_1_interp.txt
 0          ips 0: no PostScript file, 1: PS file, 2: EPS, name follows (30 char.)
pippo.eps
 1, 321, 10 ifirst_step,itot,increm (output time steps)
 1          1: phi, 2: psi
 0.         phiplot: toroidal angle for (r,theta) plot only
    1       ipl0 first plot
  321       ipl1 last  plot
  0,0       l-min, l-max Fourier components used (0,0 all)
    1       iflag_rtheta, 1 plot in (r,theta,phiplot) plane (ph/ap_xxxx.gif)
      0         iflag_test, superimposes to (r,theta,phtest) plot the
                  i-th test particle (i=iflag_test)
      0         iflag_trajectory, plot particle trajectory in
                  (rtest,thtest,phtest,utest) space
                  (trajec_[i]_xxxx.gif, trajRZ_[i]_xxxx.gif, trajXY_[i]_xxxx.gif)
    0       iflag_fourier_comp: 1 plot fourier component profiles
      0         iflag_contour: 1 2D plot of Fourier components (mn_ph/ap_xxxx.gif),
                               2 contour plot (mn_C_ph/ap_xxxx.gif),
                               0 do both
    0       iflag_power_spectrum: 1 power power spectrum of field in the
                                    plane (omega,r)
      576.      time window for Fourier transform
      0. 0.     r0, r1 (min, max in r,     if 0. 0. use max available interval)
      0. 0.25   w0, w1 (min, max in omega, if 0. 0. use max available interval)
      1         ihann- Hanning function 0: off, 1: on
      3         ibuffer (1: no buffer, n>1: zero buffer n-1 times)
      0  0.001  ilog, fac_zmin (color scale, 0: linear, 1: log, min value plotted)
      505  505  ndivx, ndivy for power spectrum plot axes
      .true.    logic_fill (false: only contour, true: fill)
      0         call cerca_massimi for plotta_max (0: no, 1: yes)
    0       iflag_deltate, synthetic diagnostic Delta_Te
      -0.1 0.1  csi0, csi1         (min, max in csi,
                                    if 0. 0. use max available interval)
     -50. 50.0  deltate0, deltate1 (min, max in deltate,
                                    if 0. 0. use max available interval)
 5
```

Note that the previous input file will produce a sequence of plots starting from $t\omega_{A0} = 0.$ to $t\omega_{A0} = 96.$; only a single frame will be shown in the following Figures 8, 9, 10. `iflag_test` can vary from “0” (no test particle plots) to `NPTEST` (producing plots for the i-th test particle). `iflag_trajectory=1` will produce, in addition, the trajectories of the selected test particle ($r_{\text{test}}(t)$, $\theta_{\text{test}}(t)$, $\varphi_{\text{test}}(t)$, $u_{\text{test}}(t)$).

The input file `Te_vs_erg_DIII_D_1_interp.txt` contains the vectors $r, T_e(r)$ on the `NRL` mesh for synthetic diagnostics purposes (to be produced by the user).

Note that for producing the power spectrum in the plane (r, ω) one has to choose the time window used in the FFT (`twindow` $= T_{\text{FFT}}\omega_{A0}$); the minimum frequency ω_{min} resolved is given by: $\omega_{min}/\omega_{A0} = 2\pi/$`twindow`, whereas the maximum frequency is given by $\omega_{max}/\omega_{A0} = \pi/\Delta t = \pi/$`(NWRITE*DT)`. To minimize the effect of having a finite time sequence, the data can be multiplied by a Hanning function, which essentially is a function picked at the middle of the time sequence and which goes to zero toward the edge. To increase the number of points in the frequency direction (but not the content of information!) a buffer of zeros can be added to the time sequence using the parameter `ibuffer`. Note also that the plotting routines interpolate the resulting FFT function in order to draw the contour plot curves: care should be taken in choosing the `twindow` parameter in order to resolve a specific mode (see Fig. 11 and Fig. 12 to compare the effect of different parameters on the produced spectra).

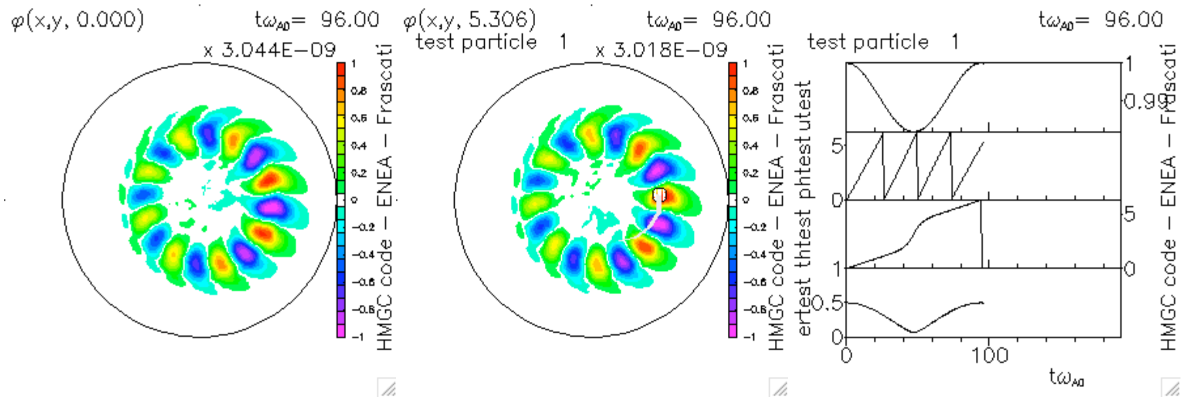

Figure 8: Frames number "161" ($t\omega_{A0} = 96.$) as produced by the program `plot_field.f` using the `xplot_field_input` data shown above. Left: $\phi(r, \theta, \varphi = 0)$. Centre: $\phi(r, \theta, \varphi)$ and superimposed the position of the first test particle (produced by assigning `iflag_test=1`). Right: trajectories of the first test particle ($r_{\rm test}$, $\theta_{\rm test}$, $\varphi_{\rm test}$, $u_{\rm test}$) (produced by assigning `iflag_test=1` and `iflag_trajectory=1`).

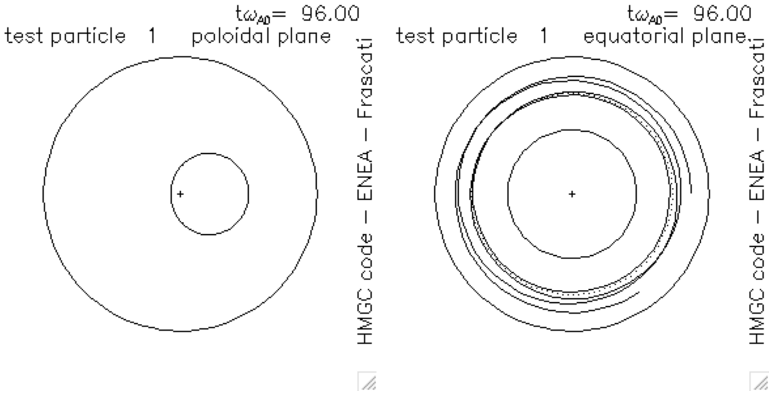

Figure 9: Test particle trajectory projected on the poloidal cross section ($\varphi = 0.$, the cross indicates $r = 0$) and on the equatorial plane ($Z = 0$, top view of the torus, dotted line refers to $r = 0$, the cross indicates the axis of symmetry of the torus) (produced by assigning `iflag_test=1` and `iflag_trajectory=1`).

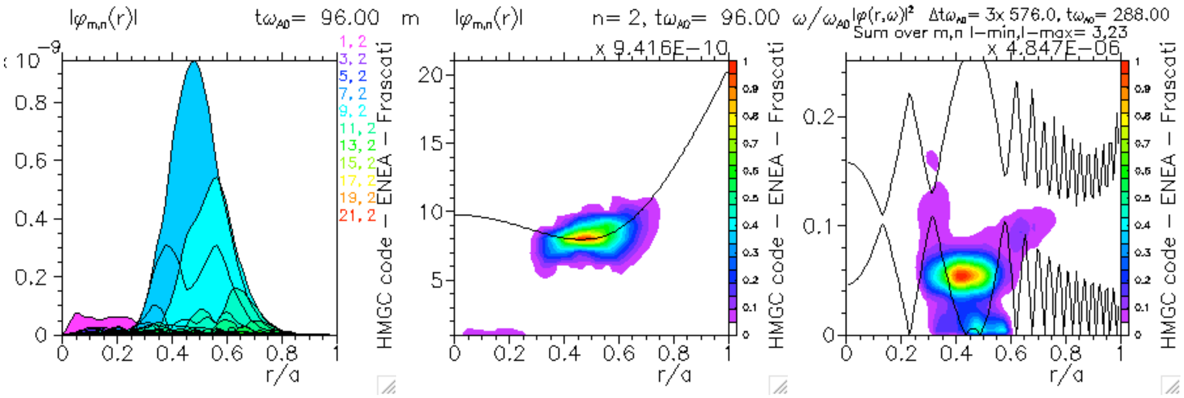

Figure 10: Left: $\phi_{m,n}(r)$ (produced by assigning `iflag_fourier_comp=1` and `iflag_contour=0`). Centre: $\phi_{m,n}(r)$ (contour plot) and superimposed the curve $m = nq(r)$. Right: frequency spectra of $\phi_{m,n}$ in the plane (r, ω) (contour plot) with superimposed the upper and lower Alfvén continua of the toroidal gap for a particular toroidal mode number (automatically chosen from the first perturbed mode, in this case $n = 2$) (produced by assigning `itot=961`, `ipl0=ipl1=481`, `iflag_power_spectrum=1` and other data from file `xplot_field_input`).

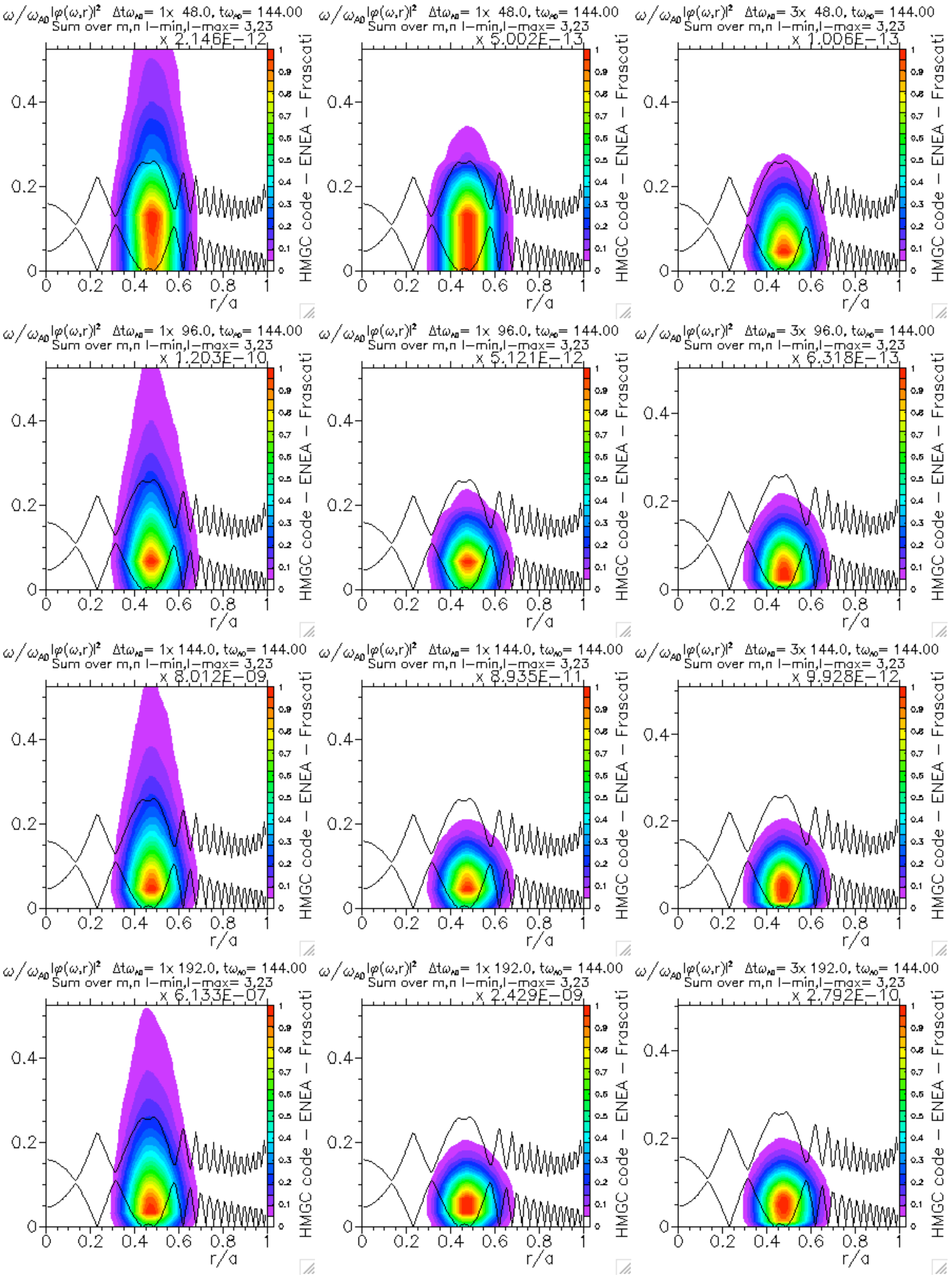

Figure 11: Frequency spectra during the linear growth phase ($t\omega_{A0}$ = 144.) and different values for `twindow` (first row: `twindow`=48., $\omega_{min}/\omega_{A0} \simeq 0.131$, second row: `twindow`=96., $\omega_{min}/\omega_{A0} \simeq 0.0654$, third row: `twindow`=144., $\omega_{min}/\omega_{A0} \simeq 0.0436$, forth row: `twindow`=192., $\omega_{min}/\omega_{A0} \simeq 0.0327$), `ihann` and `ibuffer` (first column: `ihann`=0, `ibuffer`=1, second column: `ihann`=1, `ibuffer`=1, third column: `ihann`=1, `ibuffer`=3, i.e. $\omega_{min,plot} = \omega_{min}/3$).

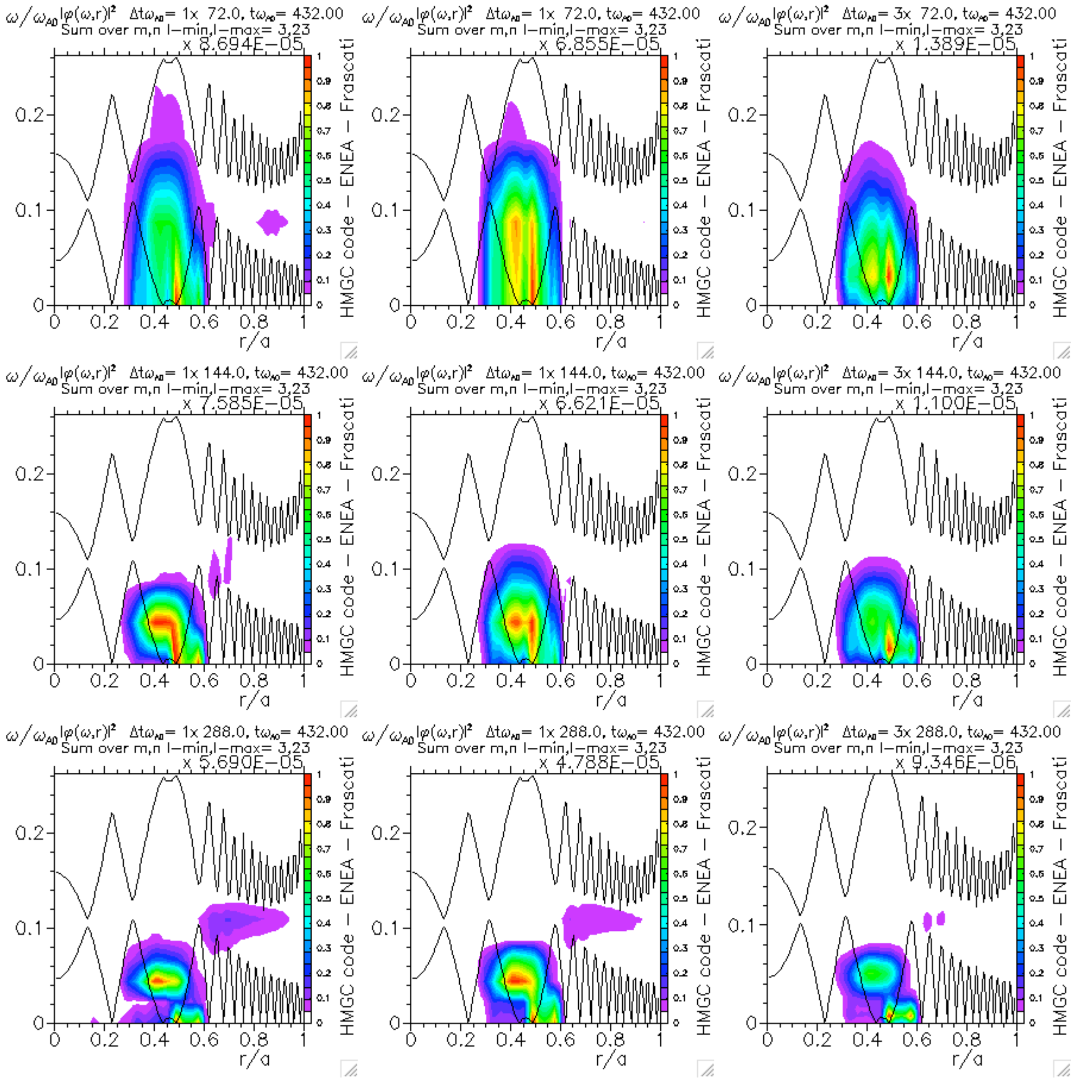

Figure 12: Frequency spectra during the saturated phase ($t\omega_{A0}$ = 432.) and different values for `twindow` (first row: `twindow`=72., $\omega_{min}/\omega_{A0} \simeq 0.0873$, second row: `twindow`=144., $\omega_{min}/\omega_{A0} \simeq 0.0436$, third row: `twindow`=288., $\omega_{min}/\omega_{A0} \simeq 0.0218$), `ihann` and `ibuffer` (first column: `ihann`=0, `ibuffer`=1, second column: `ihann`=1, `ibuffer`=1, third column: `ihann`=1, `ibuffer`=3, i.e. $\omega_{min,plot} = \omega_{min}/3$).

5.4 Output file `deltaf.data` (`deltaf_ealfa.data`)

The file `deltaf.data` contains the time series of the energetic particle distribution function $f_H(r, \mu, u)$ (thus, integrated over θ, φ), and is read by the program `plot_deltaf.f` (see Sect. 6 for the details on the distribution function). The distribution function is stored on a (r, μ, u) grid defined for plotting purposes. More precisely, the number of energetic particles $N_H(r_i, \mu_j, u_k)$ in the volume centered at (r_i, μ_j, u_k) is given by:

$$N_H(r_i, \mu_j, u_k) = n_H(0)a^3 f_H(r_i, \mu_j, u_k) dr_i d\mu_j du_k \, .$$

The file is in `unformatted` format. The file `deltaf_ealfa.data` contains the same data but on a (E, α) mesh instead of (μ, u). The data stored on the (μ, u) mesh can be transformed and plotted on a (E, α) one; note that this is not equivalent to store directly the distribution function in the (E, α) space (indeed, the single particle energy dependence on θ is lost). The data can be also summed up between to radii to reduce particle noise (see Fig. (13)). The plotting routines allow to subtract to the current $f_H(r, \mu, u, t)$ the value of $f_H(r, \mu, u, t = t_{\mathrm{relax}})$, where $t = t_{\mathrm{relax}}$ is a specific time (thus, the $\delta f_H(r, \mu, u)$ is plotted, see Fig. (14))). An example of the input file `xplot_deltaf_input` for the program `plot_deltaf.f` is listed hereafter:

```
deltaf.data
 0          ips 0: no PostScript file, 1: PS file, 2: EPS, name follows (30 char.)
pippo.eps
   1, 321,  1 ifirst_step,itot,increm (output time steps)
    1         ipl0 - first plot
  211         ipl1 - last  plot
 1            iflag - 1:plot deltaf, 2:plot deltaf_ealfa
 0            iflag_df - 0: plot f, 1: f_t-f_t_relax
 1            iflag_versus - 1: mu,u, 2: E,alpha (only if iflag=1)
 0.00 0.00    am0 am1 - (min, max in mu (E)  if 0. 0. use max available interval)
 0.00 0.00    u0, u1  - (min, max in u (alpha) if 0. 0. use max available interval)
 1            iflag_contrai - 1: sum from jer0_pl_min to jer0_pl_max
 24 40        jer0_pl_min, jer0_pl_max - (if not summed, plot jer0_pl_min)
 0            iflag_relax - 1: reset timkin_relax
 120.         timkin_relax_new
 0            iflag_renorm - 1: renormalize f_t_relax to n_H(r,t_relax)
                                and f_t to n_H(r,t) (only if  iflag_df=1)
 1            iflag_filtro - 1: plot of sedentary and displaced particles df
                                (only if  iflag_df=1)
 5
```

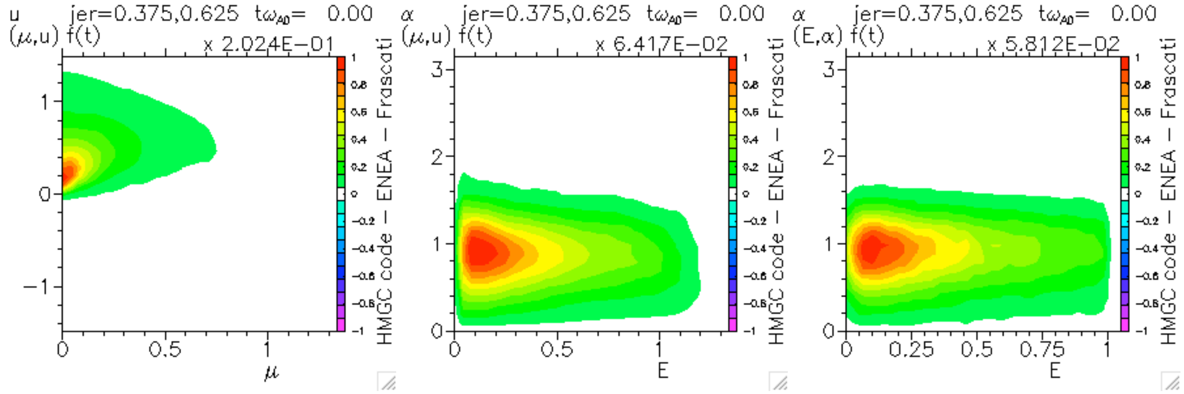

Figure 13: Left: $f_H(r,\mu,u)$. Centre: $f_H(r,\mu,u)$ transformed on the (E,α) mesh. Right: $f_H(r,E,\alpha)$. All figures are obtained summing from $r_{jer0\_pl\_min} = 0.375$ to $r_{jer0\_pl\_max} = 0.625$. Note that the input file `xplot_deltaf_input` shown in the text refers to the left plot; the figure in the centre is obtained with `iflag_versus`=2; the figure on the right is obtained reading the file `deltaf_ealfa.data`. All figures refer to the frame "1" $(t\omega_{A0} = 0.)$.

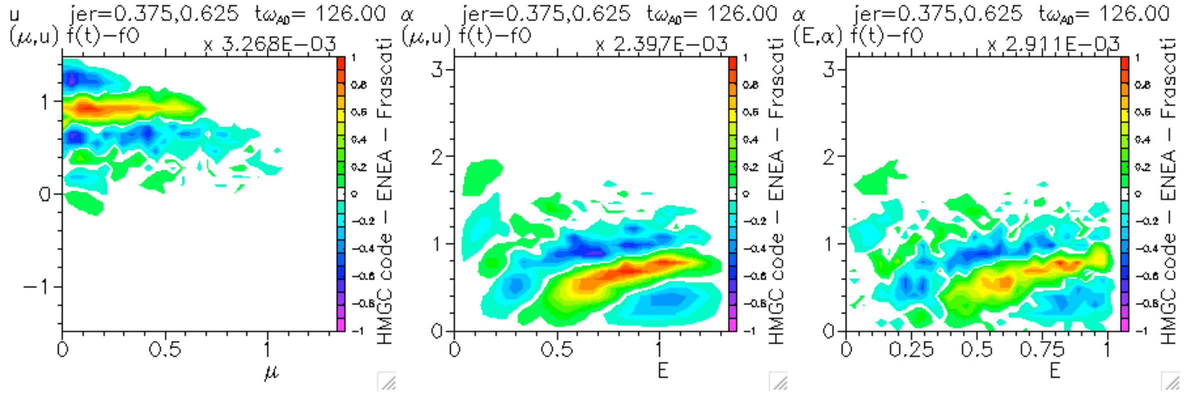

Figure 14: Left: $\delta f_H(r,\mu,u)$. Centre: $\delta f_H(r,\mu,u)$ transformed on the (E,α) mesh. Right: $\delta f_H(r,E,\alpha)$. All figures are obtained summing from $r_{jer0\_pl\_min} = 0.375$ to $r_{jer0\_pl\_max} = 0.625$. The relaxed time is $t_{\rm relax} = 120$. With respect to the input file `xplot_deltaf_input` shown in the text, the figures are obtained by assigning `iflag_df=1`. All figures refer to the frame "211" $(t\omega_{A0} = 126.)$.

5.5 Output file `power.data`

The file `power.data` contains the time series of the power exchanged between the energetic particles and the waves $P_H(r,\mu,u)$ (thus, integrated over θ, φ), and is read by the program `plot_power.f`. The power is stored on a (r,μ,u) grid defined for plotting purposes. The file is in `unformatted` format. The data stored on the (μ,u) mesh can be transformed and plotted on a (E,α) one; note that this is not equivalent to store directly the distribution function in the (E,α) space (indeed, the single particle energy dependence on θ is lost). The data can be also summed up between to radii to reduce particle noise (see Fig. (15)). An example of the input file `xplot_power_input` for the program `plot_power.f` is listed hereafter:

———————————————— ... ————————————————

```
ENERGY
power.data
 0          ips 0: no PostScript file, 1: PS file, 2: EPS, name follows (30 char.)
pippo.eps
   1, 321, 10 ifirst_step,itot,increm (output time steps)
  161         ipl0 - first plot
  161         ipl1 - last  plot
 2            iflag_versus - 1: mu,u, 2: E,alpha
 0.00 0.00    am0 am1 - (min, max in mu (E)  if 0. 0. use max available interval)
 0.00 0.00    u0, u1  - (min, max in u (alpha) if 0. 0. use max available interval)
 1            iflag_contrai - 1: sum from jer0_pl_min to jer0_pl_max
 24 40        jer0_pl_min, jer0_pl_max - (if not summed, plot jer0_pl_min)
 5
```

———————————————— ... ————————————————

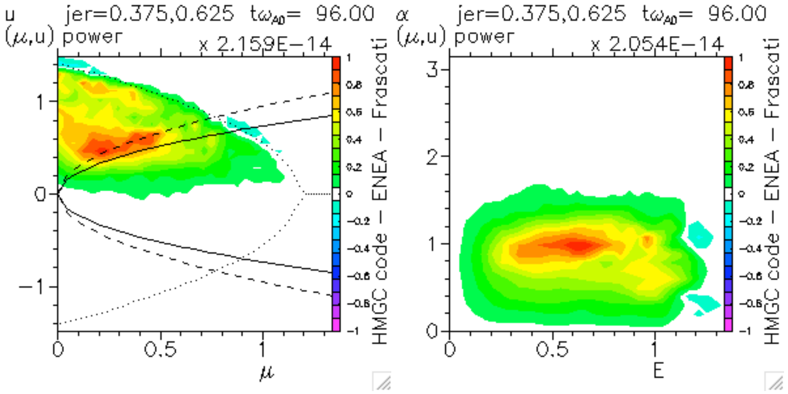

Figure 15: Left: $P_H(r,\mu,u)$. Also the curves representing the maximum energy loaded in the initial distribution function (dotted lines) and the trapped-untrapped particle boundaries for the lower (solid line) and upper (dashed line) radii considered are shown. Right: $P_H(r,E,\alpha)$. All figures are obtained summing from $r_{jer0\_pl\_min} = 0.375$ to $r_{jer0\_pl\_max} = 0.625$.

6 Energetic particle distribution functions

HMGC considers at present three different initial energetic particle equilibrium distribution functions f_{EQ}, namely anisotropic Maxwellian (`idistr=1`), anisotropic slowing down (`idistr=2`), and bi-Maxwellian (`idistr=3`). Their expressions are listed hereafter.

6.1 Maxwellian distribution function (`idistr=1`)

The Maxwellian energetic particle equilibrium distribution function $f_{\mathrm{EQ,Maxw}}$ is defined as follows:

$$
\begin{aligned}
f_{\mathrm{EQ,Maxw}} &= \frac{n_{H0}}{T_{H0}^{3/2}}\left(\frac{m_H}{2\pi}\right)^{3/2}\hat{f}_{\mathrm{EQ,Maxw}}\,, && (11)\\
\hat{f}_{\mathrm{EQ,Maxw}} &= \frac{\hat{n}(\psi)}{\tau(\psi)^{3/2}}\Theta(\alpha,\alpha_0,\Delta)e^{-E/T_H(\psi)}\,, && (12)\\
E &= \frac{1}{2}m_H u^2 + \mu\Omega_{cH}\,, && (13)\\
\Theta(\alpha,\alpha_0,\Delta) &\equiv \frac{4}{\Delta\sqrt{\pi}}\frac{\exp\left[-\left(\frac{\cos\alpha-\cos\alpha_0}{\Delta}\right)^2\right]}{\mathrm{erf}\left(\frac{1-\cos\alpha_0}{\Delta}\right)+\mathrm{erf}\left(\frac{1+\cos\alpha_0}{\Delta}\right)}\,, && (14)\\
\cos\alpha &\equiv \frac{u}{\sqrt{2E/m_H}}\,, && (15)\\
\sin^2\alpha &\equiv \frac{\mu\Omega_{cH}}{E}\,, && (16)\\
n_H(\psi) &= n_{H0}\hat{n}(\psi)\,, && (17)\\
T_H(\psi) &= T_{H0}\tau(\psi)\,, && (18)
\end{aligned}
$$

with E the energy, $n_H(\psi)$ the radial density profile, $T_H(\psi)$ the temperature, u the parallel (to the equilibrium magnetic field) velocity, μ the conserved magnetic moment, α the pitch angle of the energetic particles and $\Theta(\alpha)$ represents the anisotropy of the distribution function. $n_{H0} = n_H(r=0)$, $T_{H0} = T_H(r=0)$, $\Omega_{cH} = e_H B/(m_H c)$ with e_H, m_H the charge and the mass of the energetic particle, respectively, and B the (local) equilibrium magnetic field. The parameters entering in the anisotropy term Θ have the following meaning: α_0 is the injection angle (e.g., in the case of beams) and Δ is the width of the beam around $\cos(\alpha_0)$. In the code, the parallel velocity is normalized to the on-axis energetic particle thermal velocity $\hat{u} \equiv u/v_{th0}$, with $v_{th0} = \sqrt{T_{H0}/m_H}$, and the magnetic moment is normalized to $\hat{\mu} \equiv \mu\Omega_{cH0}/T_{H0}$, with $\Omega_{cH0} = e_H B_0/(m_H c)$.

From the slowing down definition of the distribution function, it follows that:

$$
\int_0^{\pi}\sin\alpha d\alpha\Theta(\alpha) = 2\,, \qquad (19)
$$

Relevant limits are:

$$\begin{aligned}
\Delta \longrightarrow \infty &= \Theta(\alpha) = 1 && (20)\\
\Delta \longrightarrow 0 &= \Theta(\alpha) = 2\delta(\cos\alpha - \cos\alpha_0) && (21)
\end{aligned}$$

The definition of the pressure tensor is:

$$\begin{aligned}
p_{\|} &= \int \mathrm{d}^3v\, 2F\, E \cos^2\alpha\,, && (22)\\
p_{\perp} &= \int \mathrm{d}^3v\, F\, E \sin^2\alpha\,, && (23)\\
p &= \frac{p_{\|} + 2p_{\perp}}{3} = \frac{2}{3}\int \mathrm{d}^3v\, F\, E\,, && (24)
\end{aligned}$$

where F is the distribution function. From the normalization condition (19) it follows:

$$\begin{aligned}
p_{\|} &= \frac{3}{2}p\Lambda_{\|}\,, && (25)\\
p_{\perp} &= \frac{3}{2}p\Lambda_{\perp}\,, && (26)\\
\Lambda_{\|} &= \int_0^{\pi} \mathrm{d}\alpha \sin\alpha \cos^2\alpha\, \Theta(\alpha)\,, && (27)\\
\Lambda_{\perp} &= \frac{1}{2}\int_0^{\pi} \mathrm{d}\alpha \sin^3\alpha\, \Theta(\alpha)\,, && (28)\\
\Lambda_{\perp} + \frac{1}{2}\Lambda_{\|} &= 1\,. && (29)
\end{aligned}$$

Then

$$p_{\perp}/p_{\|} = \frac{1}{\Lambda_{\|}} - \frac{1}{2}\,. \qquad (30)$$

If the ratio $p_{\perp}/p_{\|}$ is provided experimentally, the corresponding value for Δ can be obtained implicitly from Eq. (30) (see Fig. (16)). The explicit expression for $\Lambda_{\|}$ is:

$$\begin{aligned}
\Lambda_{\|} &= \int_0^{\pi} \mathrm{d}\alpha \sin\alpha \cos^2\alpha\, \Theta(\alpha) = && (31)\\
&= 2\cos^2\alpha_0 + \Delta^2 - \frac{2\Delta^2\pi^{-1/2}}{\mathrm{erf}\left(\frac{1-\cos\alpha_0}{\Delta}\right) + \mathrm{erf}\left(\frac{1+\cos\alpha_0}{\Delta}\right)} \times\\
&\left\{\left(\frac{1+\cos\alpha_0}{\Delta}\right)\exp\left[-\left(\frac{1-\cos\alpha_0}{\Delta}\right)^2\right] + \left(\frac{1-\cos\alpha_0}{\Delta}\right)\exp\left[-\left(\frac{1+\cos\alpha_0}{\Delta}\right)^2\right]\right\}.
\end{aligned}$$

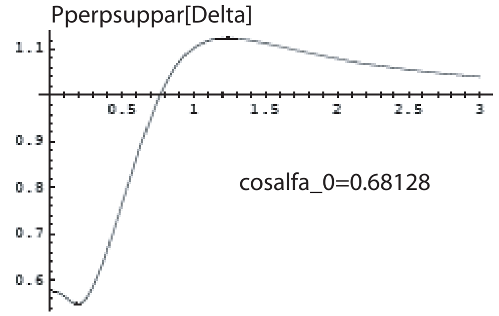

Figure 16: p_{perp}/p_{par} vs. Δ.

6.2 Slowing down distribution function (`idistr=2`)

The slowing down energetic particle equilibrium distribution function $f_{\rm EQ,sd}$ is defined as follows:

$$f_{\rm EQ,sd} = \frac{n_{H0}}{E_{crit,0}^{3/2}} \frac{3}{4\pi} \left(\frac{m_H}{2}\right)^{3/2} \hat{f}_{\rm EQ,sd}\,, \tag{32}$$

$$\hat{f}_{\rm EQ,sd} = \frac{\hat{n}(\psi)}{\tau(\psi)^{3/2}} \frac{\Theta(\alpha,\alpha_0,\Delta)}{\left[(E/E_{crit})^{3/2}+1\right]\ln\left[1+(E_0/E_{crit})^{3/2}\right]}\,, \tag{33}$$

In this case the quantity τ is given by $\tau \equiv E_{crit}(\psi)/E_{crit,0}$ with E_{crit} being the critical energy (see Stix [8]):

$$E_{\rm crit}(\psi)[{\rm MeV}] = 14.8 T_e(\psi)[{\rm MeV}] \left(\frac{A_H^{3/2}}{n_e(\psi)} \sum_i \frac{n_i(\psi) Z_i^2}{A_i}\right)^{2/3}\,, \tag{34}$$

with A_H the mass number of the energetic ions, A_i and Z_i the mass number and electric charge of the bulk ions (plus impurity, eventually) and T_e the electron temperature in MeV, n_e and n_i the electron and bulk ion densities. E_0 is the birth energy of the energetic particles and $E_{crit,0} = E_{crit}(0)$. In the code, the parallel velocity is normalized, in this case, to the energetic particle birth energy velocity $\hat{u} \equiv u/v_{th0}$, with $v_{th0} = \sqrt{E_0/m_H}$.

6.3 Bi-Maxwellian distribution function (`idistr=3`)

The Bi-Maxwellian energetic particle equilibrium distribution function $f_{\rm EQ,bi-Maxw}$ is defined as follows:

$$f_{\rm EQ,bi-Maxw} = \frac{n_{H0}}{T_{\perp H0}^{3/2}} \left(\frac{m_H}{2\pi}\right)^{3/2} \hat{f}_{\rm EQ,bi-Maxw}\,, \tag{35}$$

$$\hat{f}_{\rm EQ,bi-Maxw} = \frac{\hat{n}(\psi)}{\tau_\perp(\psi)\tau_\parallel(\psi)^{1/2}} \left(\frac{T_{\perp H0}}{T_{\parallel H0}}\right)^{1/2} \exp\left[-\left(\frac{m_H u^2}{2T_{\parallel H0}\tau_\parallel(\psi)} + \frac{\mu\Omega_{cH}}{T_{\perp H0}\tau_\perp(\psi)}\right)\right]. \tag{36}$$

$$
\begin{aligned}
T_{\perp H}(\psi) &= T_{\perp H0}\tau_{\perp}(\psi)\,, \qquad (37)\\
T_{\parallel H}(\psi) &= T_{\parallel H0}\tau_{\parallel}(\psi)\,. \qquad (38)
\end{aligned}
$$

In this case the normalized quantities are $\hat{u} \equiv u/v_{th0}$, with $v_{th0} = \sqrt{T_{\perp H0}/m_H}$ and $\hat{\mu} \equiv \mu\Omega_{cH0}/T_{\perp H0}$. The quantity σ_0 is $\sigma_0 \equiv T_{\perp H0}/T_{\parallel H0}$.

7 How to setup an HMGC run

In this section we will give indications on how to set up a specific run of HMGC: we will refer to a DIII-D discharge (#122117) analyzed in Ref. [7].

W.W. Heidbrink provide us a set of radial profiles for the following quantities ($t = 0.414$ s) of the DIII-D discharge (see Table (10) and Fig. (17)).

$n_e(r)$	Electron density (cm-3)
$n_i(r)$	(Thermal) deuterium density (cm-3)
$n_{imp}(r)$	Impurity density (carbon) (cm-3)
$n_H(r)$	Beam-ion density (cm-3) [Note that this transp value is reduced by fast-ion transport]
$T_e(r)$	Electron temperature (eV)
$\beta_H(r)$	Beam beta toroidal (dimensionless) [Note that this transp value is reduced by fast-ion transport]
$\omega_\varphi(r)$	Toroidal angular velocity (rad/s)
$q(r)$	safety factor

Table 10: Experimental radial profiles provided by DIII-D team.

The quantity $\beta_H(r)$ will not be used as an input data, but only to compare with the computed $\beta_H(r)$ from the code (this latter quantity will depend, indeed, on the model used to load the initial distribution function). The quantity $\omega_\varphi(r)$ could be used to compare the experimental frequency ($\nu_{\rm exp}$) spectra with the ones obtained by the code ($\nu_{\rm code}$): $\nu_{\rm exp} = \nu_{\rm code} + (n/2\pi)\omega_\varphi$, with n the toroidal mode number.

In Fig.(18) the geometry of the neutral beam is shown. The beam is essentially tangential, wit injection angle $\alpha_0 \equiv \arccos(R_{tan}/R_0) = \arccos(1.15\text{m}/1.688\text{m}) = 47.055\,\text{deg}$. The average birth energy is $E_0 = 0.077$ MeV ($= 2/3 \times 0.075$ MeV $+ 1/3 \times 0.081$ MeV).

From the data profiles provided and from global information a single toroidal mode number $n = 2$ simulation can be set up performing the following steps.

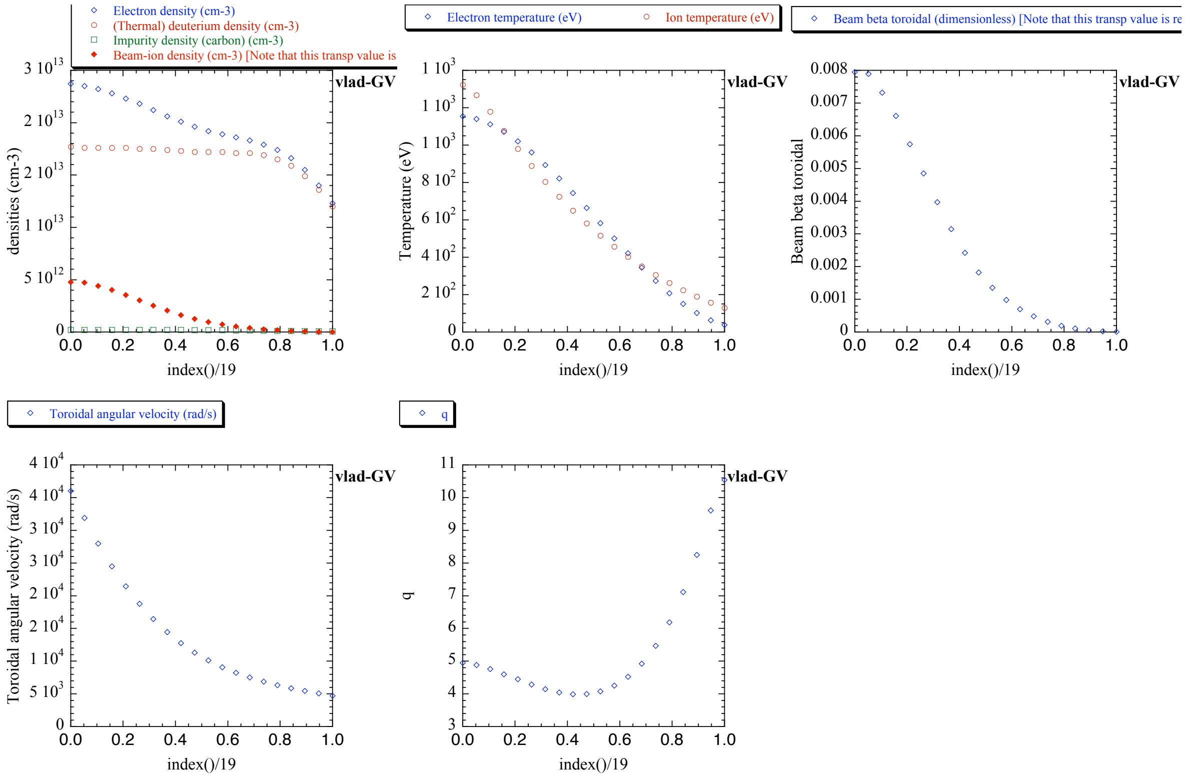

Figure 17: Experimental profiles for the DIII-D shot # 122117.

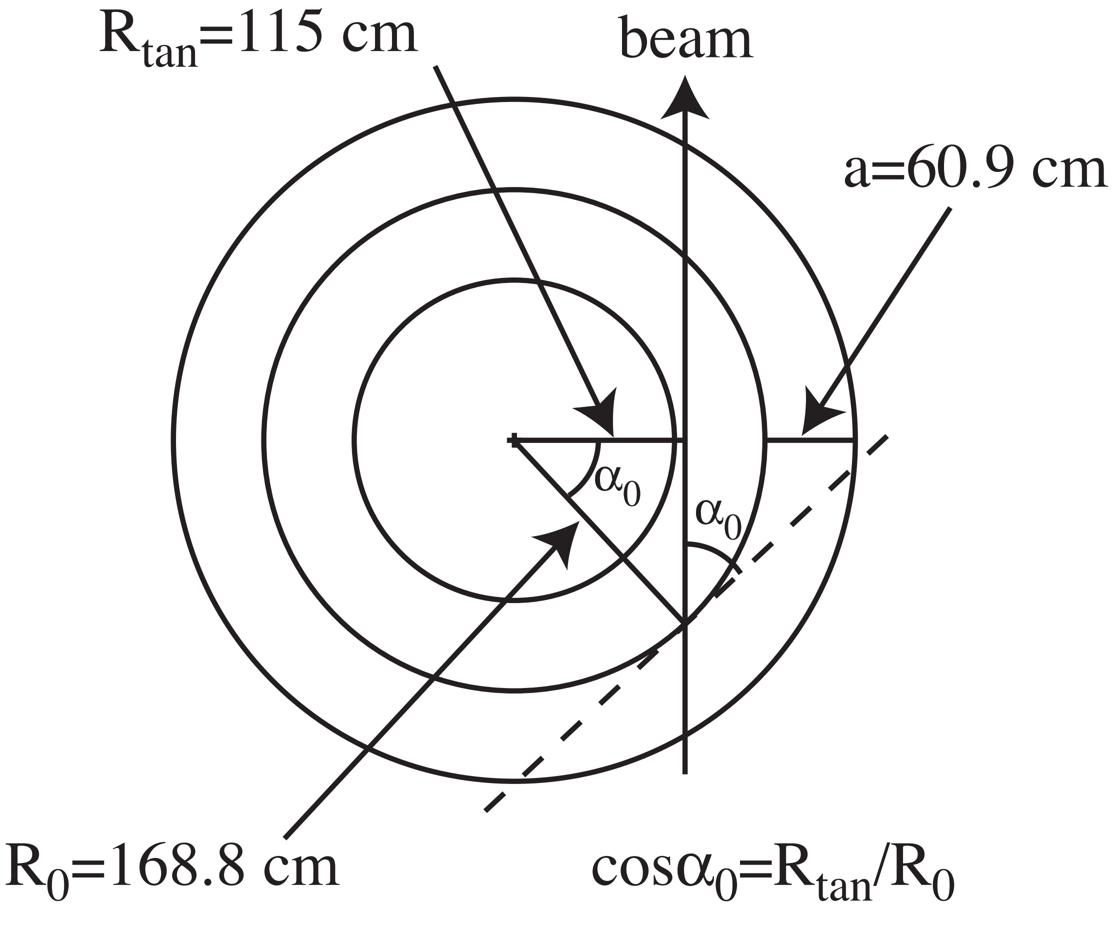

Figure 18: Beam geometry.

7.1 How to setup an HMGC run: preparing the equilibrium file EQNEW

Using the experimental $q(r)$ profile, a EQNEW file should be produced following Sect.(2).

7.2 How to setup an HMGC run: preparing the mode file (TMODE and modi_inc)

The Fourier components considered in the simulation can be chosen considering the range of the $q(r)$ profile, typically starting from $m_{min} = 1$ up to $m_{max} \approx nq(r = 1)$. For the specific case, the modes $(m, n) = (0, 0), (1, 0)$ have been included for the equilibrium, and the modes $(1, 2)$ up to $(21, 2)$ for the perturbation. ($m_{max} = 2 \times q(r = 1) = 2 \times 10.5385 \approx 21$; note that $4 \times (m_{max} - 1) = 80$ is a valid number for the ESSL FFT routine).

7.3 How to setup an HMGC run: plasma parameters (file PARAM)

The bulk plasma density normalized profile should be assigned following one of the proposed expressions (NPROFI flag, see Table (3)). For representing experimental data it is usually convenient to choose NPROFI=1; the three parameters α, β, ρ_a can be obtained by fitting the experimental profile. Normalized resistivity η and normalized viscosity ν should be chosen such that numerical stability is assured: typical values used in the simulations are $\eta = 1. \times 10^{-5}$, $\nu = 1. \times 10^{-8}$. The initial amplitude perturbation AMP should be such to be well below the saturation amplitude, in order to provide a sufficiently long "linear" phase of the simulation. Actually, if is initialized only a single Fourier component (the one at the centre of the poloidal spectrum). A time step DT must be chosen: usually the MHD module loop on a shorter time step, whereas the gyrokinetic module is called every NSUBCY time steps. One should take care that the fastest particles considered in the simulations do not cross more than one cell in the θ and φ directions (or "equivalent" cell, if the nogrid version is used) to integrate correctly their motion.

7.4 How to setup an HMGC run: energetic particle profiles files den_spli.data and temp_spli.data

The integration of the $r/q(r)$ profile will give a $\psi(r)$ mesh (see eq. (9) in Sect. (3.8)).

The $E_{crit}(r)/E_{crit,0}$ normalized profile can be derived from the radial profiles of the above listed plasma quantities (e.g., using a spreadsheet) and it will be used as temp_exp_DIII_D_1 file (table of $\psi(r)$,$E_{crit}(r)/E_{crit,0}$).

Similarly, the energetic particle normalized density profile will be derived by the "Beam-ion density profile" and used as `den_exp_DIII_D_1` file (table of $\psi(r)$,$n_H(r)/n_{H0}$).

For each of the previous files, the program `interp_spline.f` is called by the execution script, producing the files `den_spli.data` and `temp_spli.data`.

7.5 How to setup an HMGC run: energetic particle dimensioning (file grid_inc)

The parameters which describe the energetic particle dimensioning have been chosen according to Table (11).

`NLR`	64	low n, should be such to resolve the typical radial width structure of the perturbed mode
`NTH`	168	usually take 8 poloidal mesh points times m_{max}
`nintphi`	$2 \times (n-1)$	
`nph_su_nintphi`	4	
`ippc`	2	two particle per cell per each direction (r, θ, φ)
`nne`	672	as obtained by the program `calcolo_nne.f`
`NMODOM`	27	see Table (5)
`NRZ`	5	it depends from the Shafranov shift of the equilibrium (see output of the program `eqe3aaab.f`, typically `NRZ` such to include the magnetic axis is taken).

Table 11: Preparing the `grid_inc` file.

7.6 How to setup an HMGC run: energetic particle parameters (file KININP)

The energetic particle parameters can be calculated from the experimental data provided, namely, energetic ion specie (D: $A_H = 2$, $Z_H = 1$), toroidal field $B_{T0} = 2$ T, $R_0 = 1.688$ m, $a = 0.609$ m, Carbon impurity, $E_0 = 0.077$ MeV, volume averaged value of $p_{par}/p_{perp} \approx 1.44$ for the beam ions.

`RHOSA`	ρ_{H0}/a	0.032863457d0	$1.02 \times 10^{-1} A_H^{1/2} Z_H^{-1} E_{0,\mathrm{MeV}}^{1/2} B_{T0,\mathrm{Tesla}}^{-1} a_{\mathrm{m}}^{-1}$
		v_{th}	$9.79 \times 10^{8} E_{0,\mathrm{MeV}}^{1/2} A_H^{-1/2}$
		v_{A0}	$2.18 \times 10^{8} B_{T0,\mathrm{Tesla}} \times$ $(A_i n_{i,10^{20}\times\mathrm{m}^{-3}} + A_{imp} n_{imp,10^{20}\times\mathrm{m}^{-3}})^{-1/2}$
`VTHSVA`	v_{th}/v_{A0}	0.271063836d0	$4.491 E_{0,\mathrm{MeV}}^{1/2} m_H^{-1/2} B_{T0,\mathrm{Tesla}}^{-1} \times$ $(A_i n_{i,10^{20}\times\mathrm{m}^{-3}} + A_{imp} n_{imp,10^{20}\times\mathrm{m}^{-3}})^{1/2}$
`usdelta_0_input`	$1/\Delta$	2.3256d0	The corresponding value of Δ has been chosen in order to match the experimental value of the ratio $p_{par}/p_{perp} \approx 1.44$
`cosalfa_0_input`	$\cos \alpha_0$	0.68128d0	$R_{\mathrm{tan}}/R_0 = 1.15\mathrm{m}/1.688\mathrm{m}$
`e0sec0`	$E_0/E_{crit,0}$	4.153850158d0	$E_{crit,0}$ from Eq. (34), evaluated at $r = 0$
`ENHSNI`	n_{H0}/n_{i0}	0.264848976d0	$n_{H0}/(n_i + n_{imp})$
`EMHSMI`	m_H/m_i	1.d0	
`IDISTR`		2	slowing down distribution function loaded

Table 12: Preparing the `KININP` file.

8 HMGC directories structure

- `~/HMGC`: main root directory:

  1. `~/HMGC/script`: directory containing compilation and execution scripts;

     (a) `xe3_HMGC`: execution script of `e3_complete.F`;

     (b) `xequil_HMGC`: execution script of `eqe3aaab.u`;

     (c) `xplot_HMGC`: execution script for post-processing plot programs (e.g., `epe3ak31`, `profilk`, `plot_density`, `plot_field`, `plot_deltaf`, `plot_power`);

     (d) `xpsi_from_rho_q_exp`: execution script of `psi_from_rho_q_exp.f`.

  2. `~/HMGC/sources`: directory containing HMGC sources;

     (a) `eqe3aaab.u`: equilibrium module;

     (b) `e3_complete.F`: MHD module;

     (c) `kin_uncomplete.F`: GK module;

     (d) `extr_push_complete.F`: pushing routine of GK module;

     (e) `extr_pressure_complete.F`: pressure routine of GK module;

     (f) commons for HMGC: `commr31_input`, `commr31_uncomplete.F`, `double`;

     (g) `calcolo_nne.f`: program to initialize the particles in the (E, α) space;

     (h) `interp_spline.f`: spline program to interpolate experimental data on the HMGC mesh;

     (i) `nlr_interp_spline`: common for the spline program `interp_spline.f`;

     (j) `psi_from_rho_q_exp.f`: program to generate $\psi(r)$ from $r, q(r)$ experimental data;

     (k) `upda.f`: simple pre-processing program (to produce `goofy.f` programs from `goofy.u`);

     (l) `upda.ksh`: shell script to run upda;

     (m) `pwr5_version`: temporary directory (architecture dependent) for producing the executable;

     (n) `makefile`: makefile;

  3. `~/HMGC/graphics`: directory containing graphics post-processor programs:

     (a) `epe3ak31.f`;

     (b) `profilk.f`;

     (c) `plot_density.f`;

     (d) `plot_energy.f`;

   (e) `plot_field.f`;
   (f) `plot_deltaf.f`;
   (g) `plot_power.f`;
   (h) `plotta_max1.f`;
   (i) `plot_equil.f`: simple plotting program to compare a computed equilibrium with experimental data;
   (j) `GVGRAPH0LIB`: directory containing some graphical routines (library `libgvgraph0.a`);
   (k) `GVGRAPH1LIB`: directory containing some graphical routines (library `libgvgraph1.a`).

- `~/HMGC_INPUTS`

  1. `~/HMGC_INPUTS/equilibrium`: directory containing equilibrium files;
  2. `~/HMGC_INPUTS/profiles`: directory containing energetic particle experimental density and temperatures profiles;

- `~/HMGC_RESULTS`: directory containing output results:

  1. `~/HMGC_RESULTS/caso_equil_DIII_D_1`: directory containing a DIII-D equilibrium test case;
  2. `~/HMGC_RESULTS/caso_n2_DIII_D_1_test_1`: directory containing a DIII-D test case;

9 Generalities on HMGC

. . .

Beginning of excerpt form the Appendices of Ref. [9]. For the references to equations not resolved please refer to the original paper.

. . .

9.1 MHD equations

We wish to start the discussion of the general properties of the Alfvén-wave spectrum using simple-model equations, the so-called ideal magnetohydrodynamic (MHD) equations, which in Gaussian units read [10]:

$$
\begin{aligned}
\frac{\partial \varrho}{\partial t} + \nabla \cdot (\varrho \mathbf{v}) &= 0\,, &(39)\\
\varrho \frac{d\mathbf{v}}{dt} &= -\nabla P + \frac{1}{c}\mathbf{J} \times \mathbf{B}\,, &(40)\\
\frac{d}{dt}\left(\frac{P}{\varrho^{\Gamma}}\right) &= 0\,, &(41)\\
\mathbf{E} + \frac{1}{c}\mathbf{v} \times \mathbf{B} &= 0\,, &(42)\\
\nabla \times \mathbf{E} &= -\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t}\,, &(43)\\
\nabla \times \mathbf{B} &= \frac{4\pi}{c}\mathbf{J}\,, &(44)\\
\nabla \cdot \mathbf{B} &= 0\,. &(45)
\end{aligned}
$$

In the above equations $\mathbf{v}$ is the fluid velocity, $\mathbf{J}$ is the plasma current, $\mathbf{B}$ is the magnetic field, ϱ is the mass density, P is the scalar pressure of the plasma, Γ is the ratio of the specific heats, c is the speed of light, and

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \tag{46}$$

is the convective derivative.

The ideal MHD equations describe the plasma as a single fluid. In particular, eq. (39) describes the time evolution of mass (conservation of the total number of particles). Equation (40) describes the time evolution of momentum, showing that the fluid is subject to inertial, pressure-gradient and magnetic forces. Equation (41) is the equation of state and generally describes the polytropic evolution of the plasma. It may be combined with the continuity equation and written as

$$\frac{dP}{dt} = -\Gamma P \nabla \cdot \mathbf{v}\,. \tag{47}$$

Equation (42), the so-called *ideal* Ohm's law, describes the plasma as a perfectly conducting fluid (from which the expression "*ideal* MHD" originates). Note that in the more general case in which plasma resistivity η is considered, eq. (42) is replaced by

$$\mathbf{E} + \frac{1}{c}\mathbf{v} \times \mathbf{B} = \eta \mathbf{J} \,. \tag{48}$$

———————————————— ... ————————————————

The elliptic operator Δ^* is given by

$$\Delta^* \psi = R^2 \nabla \cdot \left(\frac{\nabla \psi}{R^2} \right) = R \frac{\partial}{\partial R} \left(\frac{1}{R} \frac{\partial \psi}{\partial R} \right) + \frac{\partial^2 \psi}{\partial Z^2} \,. \tag{49}$$

———————————————— ... ————————————————

9.2 Order-ϵ^3 reduced MHD

We introduce, here, a simplified version of the resistive MHD equations, which will be useful later for the non-linear study of the Alfvén modes, greatly reducing the complexity of the problem. We start from the resistive MHD equations (39-45), with the Ohm's law generalized by eq. (48).

Since tokamak plasmas are characterized by values of the safety factor $q(r) \approx (rB_\varphi)/(RB_\vartheta) \approx O(1)$ (B_φ and B_ϑ are, respectively, the toroidal and poloidal component of the magnetic field) and inverse aspect ratio $\epsilon = a/R_0$ much lower than unity, MHD equations can be simplified by expanding in powers of ϵ. This procedure has been widely used, since the first paper of Strauss [11], both for analytical and numerical work. At the leading order in ϵ, $O(\epsilon^2)$, and considering the low-β approximation, $\beta \approx O(\epsilon^2)$, the reduced-MHD equations describe the plasma in the cylindrical approximation. Toroidal corrections enter the equations at the next order in the inverse aspect ratio expansion. The derivation of these equations has been described in detail in ref. [12] and is only briefly reported here.

Following the low-β tokamak ordering, it is possible to write

$$\frac{v_\perp}{v_A} \approx \frac{B_\perp}{B_\varphi} \approx \frac{\mathbf{B}/B \cdot \nabla}{\nabla_\perp} \approx O(\epsilon) \,,$$

$$\frac{v_\varphi}{v_A} \approx \frac{\nabla \cdot \mathbf{v}_\perp}{v_A/a} \approx \frac{\nabla (RB_\varphi)}{B_\varphi} \approx O(\epsilon^2) \,, \qquad \frac{\partial}{\partial t} \approx \frac{v_A}{R} \,.$$

Here, a cylindrical-coordinate system (R, Z, φ) has been used, and the subscript $\perp$ denotes components perpendicular to $\nabla \varphi$. The magnetic field can be written as

$$\mathbf{B} = \left(F_0 + \tilde{F} \right) \nabla \varphi + R_0 \nabla \psi \times \nabla \varphi + O(\epsilon^3 B_\varphi) \,, \tag{50}$$

where ψ is the poloidal magnetic flux function, $F_0 = R_0 B_0$, B_0 is the vacuum (toroidal) magnetic field at $R = R_0$, and $\tilde{F} \approx O(\epsilon^2 F_0)$ is given, at the leading order, by equilibrium corrections. Substituting eq. (50) and Ohm's law, eq. (48), into Faraday's law, eq. (43), we obtain

$$R_0\frac{\partial\psi}{\partial t}\nabla\varphi + c\left(\eta\mathbf{J} - \frac{1}{c}\mathbf{v}\times\mathbf{B}\right) = -c\nabla\phi + O(\epsilon^4 v_A B_\varphi)\,, \tag{51}$$

where ϕ is the scalar potential. Taking the cross product by $\nabla\varphi$, eq. (51) can be solved with respect to $\mathbf{v}_\perp$:

$$\mathbf{v}_\perp = -\frac{cR^2}{R_0 B_0}\nabla\phi\times\nabla\varphi + O(\epsilon^3 v_A)\,. \tag{52}$$

Equation (52) states that, at the lowest order, the perpendicular velocity is given by the $\mathbf{E}\times\mathbf{B}$ drift. Then, taking the $\nabla\varphi$ component of eq. (51), the following equation for the evolution of the poloidal magnetic flux function is obtained:

$$\frac{\partial\psi}{\partial t} = -\frac{cR^2}{R_0 B_0}\nabla\psi\times\nabla\varphi\cdot\nabla\phi - \frac{c}{R_0}\frac{\partial\phi}{\partial\varphi} + \eta\frac{c^2}{4\pi}\Delta^*\psi + O(\epsilon^4 v_A B_\varphi)\,, \tag{53}$$

with the Grad-Shafranov operator Δ^* defined by eq. (49).

Upon applying the operator $\nabla\varphi\cdot\nabla\times R^2\ldots$ to the momentum equation, eq. (40), the following equation for the evolution of the scalar potential is obtained:

$$\begin{aligned}\hat{\varrho}\left(\frac{D}{Dt} - \frac{2c}{R_0 B_0}\frac{\partial\phi}{\partial Z}\right)\nabla_\perp^2\phi + \nabla\hat{\varrho}\cdot\left(\frac{D}{Dt} - \frac{c}{R_0 B_0}\frac{\partial\phi}{\partial Z}\right)\nabla\phi = \\ -\frac{B_0}{4\pi c}\mathbf{B}\cdot\nabla\Delta^*\psi - \frac{B_0}{cR_0}\nabla\cdot\left(R^2\nabla P\times\nabla\varphi\right) + O(\epsilon^4\varrho\frac{v_A^2 B_\varphi}{a^2 c})\,,\end{aligned} \tag{54}$$

where

$$\hat{\varrho} = \frac{R^2}{R_0^2}\varrho\,, \qquad \frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v}_\perp\cdot\nabla\,,$$

$$\nabla_\perp^2 \equiv \frac{1}{R}\frac{\partial}{\partial R}R\frac{\partial}{\partial R} + \frac{\partial^2}{\partial Z^2}\,.$$

Note that, both in eq. (53) and eq. (54), v_φ and $\tilde{F}$ enter only at the fourth order in ϵ. In eq. (54) the dependence on the density gradient has been retained explicitly. With the particular choice of the mass density $\varrho R^2 = \hat{\varrho}R_0^2 = const$, and using the definition of $\mathbf{v}_\perp$ given in eq. (52), the continuity equation, eq. (39), is satisfied up to the third order. The pressure equation becomes

$$\frac{DP}{Dt} = O(\epsilon^4\frac{v_A B_\varphi^2}{a})\,. \tag{55}$$

———————— . . . ————————

9.3 Hybrid MHD-kinetic models

In order to include in this model the effects on an energetic-ion population, we can take advantage from the fact that the energetic particle density is typically much smaller than the bulk plasma density. The following ordering can then be adopted:

$$\frac{n_H}{n_i} \approx O(\epsilon^3)\,, \quad \frac{T_H}{T_i} \approx O(\epsilon^{-2})\,,$$

where n_H (n_i) and T_H (T_i) are the energetic particle (bulk ion) density and temperature respectively. Thus, the following ordering for the ratio of the energetic to bulk ion beta follows:

$$\frac{\beta_H}{\beta_i} \approx O(\epsilon)\,,$$

It can be shown [13] that, making use of the above ordering, the MHD momentum equation is modified by a term which represent the perpendicular component of the divergence of the energetic-particle stress tensor $\mathbf{\Pi}_H$ (in ref. [13] an alternative, equivalent form in which the electric current associated to the energetic ions appears, instead of the energetic-particle stress tensor, is also derived). Thus, the $O(\epsilon^3)$ equation for the evolution of the scalar potential is modified as follows

$$\begin{aligned}\hat{\varrho}\left(\frac{D}{Dt}-\frac{2c}{R_0B_0}\frac{\partial\phi}{\partial Z}\right)\nabla_\perp^2\phi+\nabla\hat{\varrho}\cdot\left(\frac{D}{Dt}-\frac{c}{R_0B_0}\frac{\partial\phi}{\partial Z}\right)\nabla\phi = \\ -\frac{B_0}{4\pi c}\mathbf{B}\cdot\nabla\Delta^*\psi-\frac{B_0}{cR_0}\nabla\cdot\left[R^2\left(\nabla P+\nabla\cdot\mathbf{\Pi}_H\right)\times\nabla\varphi\right]+O(\epsilon^4\varrho\frac{v_A^2B_\varphi}{a^2c})\,. \qquad (56)\end{aligned}$$

In order to close the set of reduced MHD eqs. (53) and (56), the hot-particle stress-tensor components can be evaluated by directly calculating the appropriate velocity moment of the distribution function for the particle population moving in the perturbed fields ψ and ϕ (see appendix sect. 9.4).

——————————— · · · ———————————

9.4 Hybrid MHD-kinetic code

In this section, we describe the code that solves the $O(\epsilon^3)$ reduced MHD model, in the limit of zero bulk-plasma pressure. In such limit, only eqs. (53) and (56) need to be solved. As a boundary condition, we take a rigid conducting wall at the plasma edge. The numerical tool [14, 15, 16] used to solve the $O(\epsilon^3)$ model is based on a field solver originating from an existing $O(\epsilon^2)$ code [17]. Such field solver uses toroidal coordinates (r, ϑ, φ), finite differences in the radial direction (r) and Fourier expansion in the poloidal (ϑ) and toroidal (φ) directions. The coupled equations for the Fourier components of the magnetic and velocity stream functions ψ and ϕ are advanced in time using a semi-implicit algorithm, where all the linear terms that couple with the cylindrical part of the equilibrium (*i.e.*, the component having poloidal and toroidal mode numbers $(m,n) = (0,0)$) are treated implicitly. The non-linear terms, the terms which arise from the toroidal corrections to the cylindrical approximation and the contributions of the energetic particles (the term containing $\nabla \cdot \mathbf{\Pi}_H$ in eq. (56)) are treated explicitly. Moreover, only the Fourier components in a half plane of the (m,n) space are evolved, the ones that fall in the other half plane being recovered from the reality condition of the solution:

$$\hat{\psi}_{-m,-n}(r,t) = \hat{\psi}^*_{m,n}(r,t)\,, \quad \hat{\phi}_{-m,-n}(r,t) = \hat{\phi}^*_{m,n}(r,t)\,. \tag{57}$$

The equilibrium configuration used for numerical simulations can be exactly calculated to the desired order in ϵ, starting from the expression for the equilibrium toroidal current

$$\Delta^* \psi^{eq} = -\frac{4\pi}{c}\frac{R}{R_0} J_{0\varphi}\,, \tag{58}$$

and expanding ψ^{eq} in powers of ϵ,

$$\psi^{eq}(r,\vartheta) = \psi_0^{eq}(r) + \psi_1^{eq}(r,\vartheta) + O(\epsilon^2 \psi_0^{eq})\,.$$

In the toroidal coordinate system (r,ϑ,φ) the Grad-Shafranov operator can be expressed as

$$\Delta^* = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \vartheta^2} - \frac{1}{R}\left(\cos\vartheta\frac{\partial}{\partial r} - \frac{\sin\vartheta}{r}\frac{\partial}{\partial \vartheta}\right)\,,$$

with $R = R_0 + r\cos\vartheta$. To the leading order, that is in the cylindrical approximation, eq. (58) is given by

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{d\psi_0^{eq}}{dr}\right) = -\frac{4\pi}{c} J_{0\varphi}[\psi_0^{eq}(r)]\,, \tag{59}$$

yielding

$$\frac{d\psi_0^{eq}}{dr} = -\frac{B_0}{R_0}\frac{r}{q(r)}\,,$$

which can be integrated assigning $\psi_0^{eq}(a) = 0$ and $q(r)$, the safety factor in the cylindrical ($O(\epsilon^2)$) approximation ($q(r) = rB_{0\varphi}/(R_0 B_{0\vartheta})$). Equation (59) gives the symmetric ($m = 0, n = 0$) Fourier component of the poloidal magnetic flux function ψ^{eq}. To the next order in ϵ, eq. (58) yields

$$\begin{aligned}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\psi_1^{eq}}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\psi_1^{eq}}{\partial\vartheta^2} - \frac{1}{R_0}\cos\vartheta\frac{d\psi_0^{eq}}{dr} = \\ -\frac{4\pi}{c}\psi_1^{eq}\frac{d}{d\psi^{eq}}\left(\frac{R}{R_0}J_{0\varphi}\right)\bigg|_{\psi^{eq}=\psi_0^{eq}}\,.\end{aligned} \tag{60}$$

Equation (60) admits solutions of the form

$$\psi_1^{eq}(r,\vartheta) = \psi_1^{eq}(r)\cos\vartheta = \Delta(r)\cos\vartheta\frac{d\psi_0^{eq}}{dr}\,, \tag{61}$$

where we have introduced the so-called Shafranov shift Δ. Substituting eq. (61) into eq. (60) and using the leading order solution of eq. (59), the following equation for the Shafranov shift Δ is obtained:

$$\frac{1}{r}\frac{d}{dr}\left[r\left(\frac{d\psi_0^{eq}}{dr}\right)^2\frac{d\Delta}{dr}\right]-\frac{1}{R_0}\left(\frac{d\psi_0^{eq}}{dr}\right)^2=0\,. \tag{62}$$

Equation (62) can be integrated assigning the radial derivative of the Shafranov shift at the center $\Delta'(0)=0$ (regularity condition) and $\Delta(a)=0$ (corresponding to $\psi^{eq}=0$ on the rigid conducting wall), to obtain $\Delta(r)$. The substitution of $\Delta(r)$ into eq. (61) allows us to obtain the first-order $(1,0)$ Fourier component of the magnetic flux function ψ^{eq}, thus completing the equilibrium solution at the desired order. Note that, once fixed r and ϑ, the quantity $\Delta(r)$ corresponds to the shift, with respect to the center of the poloidal cross section, of the geometric center of the magnetic surface labelled by the value $\psi^{eq}(r,\vartheta)$. Such a shift causes shear Alfvén waves, even when propagating along the magnetic field line, to cross the radial grid, thus imposing restrictions on the time step of integration [12]. Further restrictions are imposed by the strength of the explicitly solved terms (as, *e.g.*, in the case of high inverse aspect ratio equilibria and/or highly non-linear cases).

The term $\mathbf{\Pi}_H$ in eq. (56) is the pressure-tensor of the energetic (hot) ions; it can be expressed in terms of the corresponding distribution function f_H ($\mathbf{\Pi}_H \equiv m_H\int d^3v\mathbf{vv}f_H$, with m_H being the energetic-ion mass), to be determined by solving the Vlasov equation (the collisionless limit of the Boltzmann equation). Since the time scale of the dynamics we want to analyze is long compared to a cyclotron period, it is convenient [18, 19] to solve the Vlasov equation in the gyrocenter-coordinate system $\overline{Z}\equiv\left(\overline{\mathbf{R}},\overline{M},\overline{U},\overline{\theta}\right)$, where $\overline{\mathbf{R}}$ is the gyrocenter position, $\overline{M}$ is the magnetic moment, $\overline{U}$ is the parallel velocity (*i.e.*, the velocity along the magnetic-field line), and $\overline{\theta}$ is the gyrophase. This corresponds to averaging the single-particle equations of motion over many cyclotron orbits and allows one to retain the relevant finite-Larmor-radius effects without resolving the details of the gyromotion. Such a choice is particularly suited for numerical time integration of the particle motion, as the numerical-stability constraint on the time-step size turns out to be much less severe than that we would obtain without adopting the averaging procedure.

The hot-particle pressure tensor assumes the following form, in terms of the gyrocenter-coordinate distribution function,

$$\begin{aligned}\mathbf{\Pi}_H\left(t,\mathbf{x}\right)=\frac{1}{m_H^2}\int\quad & d^6\overline{Z}D_{z_c\to\overline{Z}}\overline{F}_H\left(t,\overline{\mathbf{R}},\overline{M},\overline{U}\right)\times\\ & \left[\frac{\Omega_H\overline{M}}{m_H}\mathbf{I}+\hat{\mathbf{b}}\hat{\mathbf{b}}\left(\overline{U}^2-\frac{\Omega_H\overline{M}}{m_H}\right)\right]\delta\left(\mathbf{x}-\overline{\mathbf{R}}\right),\end{aligned} \tag{63}$$

where $\mathbf{I}$ is the identity tensor ($I_{ij}\equiv\delta_{ij}$), $\overline{F}_H\left(t,\overline{\mathbf{R}},\overline{M},\overline{U}\right)$ is the energetic-particle distribution function in gyrocenter coordinates, and $D_{z_c\to\overline{Z}}$ is the Jacobian of the transformation from canonical to gyrocenter coordinates.

The Vlasov equation can be written as

$$\frac{d\overline{F}_H}{dt} = 0, \tag{64}$$

with

$$\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \frac{d\overline{Z}^i}{dt}\frac{\partial}{\partial \overline{Z}^i},$$

and $d\overline{Z}^i/dt$ given by the following equations of motion [14, 20, 21, 22]

$$\begin{aligned}
\frac{d\overline{\mathbf{R}}}{dt} &= \overline{U}\hat{\mathbf{b}} + \frac{e_H}{m_H\Omega_H}\hat{\mathbf{b}}\times\nabla\phi - \frac{\overline{U}}{m_H\Omega_H}\hat{\mathbf{b}}\times\nabla a_\parallel + \\
&\quad \left[\frac{\overline{M}}{m_H} + \frac{\overline{U}}{\Omega_H}\left(\overline{U} + \frac{a_\parallel}{m_H}\right)\right]\hat{\mathbf{b}}\times\nabla\ln B, \\
\frac{d\overline{M}}{dt} &= 0, \\
\frac{d\overline{U}}{dt} &= \frac{1}{m_H}\hat{\mathbf{b}}\cdot\left\{\left[\frac{e_H}{\Omega_H}\left(\overline{U} + \frac{a_\parallel}{m_H}\right)\nabla\phi + \frac{\overline{M}}{m_H}\nabla a_\parallel\right]\times\nabla\ln B + \right. \\
&\quad \left.\frac{e_H}{m_H\Omega_H}\nabla a_\parallel\times\nabla\phi\right\} - \frac{\Omega_H\overline{M}}{m_H}\hat{\mathbf{b}}\cdot\nabla\ln B.
\end{aligned} \tag{65}$$

Here, e_H and $\Omega_H \equiv e_H B/(m_H c)$ are, respectively, the energetic-ion charge and Larmor frequency. The fluctuating potential $a_\parallel$ is related to the poloidal magnetic flux function ψ through the relationship

$$a_\parallel = \frac{e_H}{c}\frac{R_0}{R}\psi. \tag{66}$$

Note that the magnetic moment $\overline{M}$ is exactly conserved in this coordinate system and that, correspondingly, neither $\overline{F}_H$ nor the equations of motion contain any dependence on the gyrophase $\overline{\theta}$.

The particle-simulation approach to the solution of Vlasov equation, eq. (64), consists in representing any phase-space function $G(t,\overline{Z})$ by its discretized form,

$$G\left(t,\overline{Z}\right) \equiv \int d^6\overline{Z}' G\left(t,\overline{Z}'\right)\,\delta\left(\overline{Z}-\overline{Z}'\right) \approx \sum_l \Delta_l\, G\left(t,\overline{Z}_l\right)\delta\left(\overline{Z}-\overline{Z}_l\right), \tag{67}$$

where Δ_l is the volume element around the phase-space marker $\overline{Z}_l$, and in assuming that each marker evolves in time according to the gyrocenter equations of motion, eqs. (65). Such markers can then be interpreted as the phase-space coordinates of a set of N_{part} "particles", and $G(t,\overline{Z})$ can be approximated by

$$G\left(t,\overline{Z}\right) \approx \sum_{l=1}^{N_{part}} \Delta_l\left(t\right)\, G\left(t,\overline{Z}_l\left(t\right)\right)\delta\left(\overline{Z}-\overline{Z}_l\left(t\right)\right). \tag{68}$$

The time-variation of the volume element $\Delta_l\,(t)$ is then given by

$$\frac{d\Delta_l}{dt} = \Delta_l(t) \left(\frac{\partial}{\partial \overline{Z}^i} \frac{d\overline{Z}^i}{dt} \right)_{t,\overline{Z}_l(t)} . \tag{69}$$

For the purpose of calculating the pressure tensor components, eq.(63), it is convenient to directly represent the quantity $D_{z_c \to \overline{Z}} \overline{F}_H$ according to its discretized form

$$D_{z_c \to \overline{Z}} \left(t, \overline{Z}\right) \overline{F}_H \left(t, \overline{Z}\right) \approx \sum_{l=1}^{N_{part}} \overline{w}_l\,(t)\, \delta \left(\overline{Z} - \overline{Z}_l\,(t)\right), \tag{70}$$

with the weight factor $\overline{w}_l$ defined by

$$\overline{w}_l\,(t) \equiv \overline{\Delta}_l\, \overline{F}_H \left(t, \overline{Z}_l\,(t)\right), \tag{71}$$

and

$$\overline{\Delta}_l \equiv \Delta_l\,(t)\;\, D_{z_c \to \overline{Z}} \left(t, \overline{Z}_l\,(t)\right). \tag{72}$$

In fact, from eqs.(64), (69), and from the Liouville theorem,

$$\frac{\partial}{\partial t} D_{z_c \to \overline{Z}} + \frac{\partial}{\partial \overline{Z}^i} \left(D_{z_c \to \overline{Z}} \frac{d\overline{Z}^i}{dt} \right) = 0 \;, \tag{73}$$

it is immediate to show that

$$\frac{d\overline{\Delta}_l}{dt} = 0, \tag{74}$$

and

$$\frac{d\overline{w}_l}{dt} = 0. \tag{75}$$

At each time step, the fluctuating electromagnetic potentials are computed at the grid points of a three-dimensional toroidal domain in terms of the Fourier components yielded by the field solver. Phase-space coordinates are then evolved in the fluctuating fields, and the pressure tensor components at the grid points are updated, in order to close the MHD equations for the next time step.

Field values at each particle position are obtained by trilinear interpolation of the fields at the vertices of the cell the particle belongs to. The corresponding trilinear weight function is adopted, after pushing the particles, in order to distribute their contribution to the pressure tensor components among the vertices of the cell. Phase-space coordinates and weights for the simulation particles are initially determined in such a way to yield a prescribed (*e.g.*, Maxwellian) distribution function. Particle pushing is performed by integrating eqs. (65) by a second-order Runge-Kutta method, more accurate than the standard $O(\Delta t)$ Euler method ($O(\Delta t^2)$ is properly retained), although more time consuming. Particles that hit the wall ($r = a$) are considered lost and are not re-injected in the plasma.

It has been shown [14, 23, 24, 25, 26, 27] that, as far as regimes are considered where the distribution function can be expected to slightly depart from the equilibrium one, it is worth limiting the numerical investigation to the evolution of the perturbed part $\delta\overline{F}_H$, defined by the relationship

$$\overline{F}_H\left(t,\overline{\mathbf{R}},\overline{M},\overline{U}\right)=\overline{F}_{H0}\left(t,\overline{\mathbf{R}},\overline{M},\overline{U}\right)+\delta\overline{F}_H\left(t,\overline{\mathbf{R}},\overline{M},\overline{U}\right), \tag{76}$$

where $\overline{F}_{H0}$ is the lowest-order ("equilibrium") distribution function.

In terms of $\delta\overline{F}_H$, eq.(64) can be written in the form

$$\frac{d\delta\overline{F}_H}{dt}=\mathcal{S}, \tag{77}$$

with

$$\mathcal{S}\equiv-\frac{d\overline{F}_{H0}}{dt}.$$

Meanwhile, eq.(70) is replaced by the following one,

$$D_{z_c\to\overline{Z}}\left(t,\overline{Z}\right)\delta\overline{F}_H\left(t,\overline{Z}\right)\approx\sum_{l=1}^{N_{part}}\overline{w}_l\left(t\right)\delta\left(\overline{Z}-\overline{Z}_l\left(t\right)\right), \tag{78}$$

with

$$\overline{w}_l\left(t\right)\equiv\overline{\Delta}_l\delta\overline{F}_H\left(t,\overline{Z}_l\left(t\right)\right), \tag{79}$$

and

$$\frac{d\overline{w}_l}{dt}=\overline{\Delta}_l\mathcal{S}\left(t,\overline{Z}_l\left(t\right)\right). \tag{80}$$

Note that eq.(76) is by no means equivalent to a linearization of the Vlasov equation, since all non-linear terms are correctly retained. The decomposition of eq.(76) is useful in reducing numerical noise as long as $|\delta\overline{F}_H|\ll|\overline{F}_{H0}|$.

In the present paper, when adopting the δF approach, we take $\overline{F}_{H0}$ to be Maxwellian

$$\overline{F}_{H0}\propto n_H\left(\overline{\mathbf{R}}\right)\exp\left(-\frac{\Omega_H\overline{M}+\frac{1}{2}m_H\overline{U}^2}{T_H}\right), \tag{81}$$

where $n_H\left(\overline{\mathbf{R}}\right)$ and T_H are, respectively, the energetic-particle equilibrium density and (uniform) temperature. The r.h.s. of eq.(77) is then given by

$$\begin{aligned}\mathcal{S}\left(t,\overline{\mathbf{R}},\overline{M},\overline{U}\right) \quad = \quad & -\overline{F}_{H0}\left\{\frac{d\overline{\mathbf{R}}}{dt}\cdot\nabla\ln n_H+\frac{e_H}{T_H}\left[\frac{\overline{M}}{m_H}+\frac{\overline{U}}{\Omega_H}\left(\overline{U}+\frac{a_\parallel}{m_H}\right)\right]\hat{\mathbf{b}}\times\nabla\ln B\cdot\nabla\phi+\right. \\ & \left.\frac{e_H\overline{U}}{T_H\Omega_H m_H}\hat{\mathbf{b}}\cdot\nabla\phi\times\nabla a_\parallel\right\}.\end{aligned} \tag{82}$$

We also assume the following model for the energetic-particle equilibrium density:

$$n_H(r)=n_{H0}\,\exp\left[-\left(\frac{r^2}{L_n^2}\right)^{\alpha_n}\right], \tag{83}$$

where n_{H0} is the on-axis density.

. . .

End of excerpt form the Appendices of Ref. [9].

. . .

References

[1] Frieman E. and Chen L., 1982 *Physics of Fluids* **25** 502

[2] Lee W., 1987 *Journal of Computational Physics* **72** 243

[3] Shinohara K., et al., 2001 *Nuclear Fusion* **41** 603

[4] Shinohara K., et al., 2002 *Nuclear Fusion* **42** 942

[5] Shinohara K., et al., 2004 *Plasma Phys. Contr. Fusion* **46** S31–45

[6] G. Vlad, S. Briguglio, G. Fogaccia, and B. Di Martino. Gridless finite-size-particle plasma simulation. *Computer Physics Communication*, 134:58–77, 2001.

[7] G. Vlad, S. Briguglio, G. Fogaccia, F. Zonca, W. W. Heidbrink, and M. A. Van Zeeland. Particle simulations of Alfvén modes in reversed-shear DIII-D discharges heated by neutral beams. In *10th IAEA Technical Meeting on Energetic Particles in Magnetic Confinement Systems, Kloster Seeon 8-10 Oct. 2007*, page OT 7, Vienna, Austria, 2007. International Atomic Energy Agency.

[8] Stix T H 1972 *Plasma Phys.* **14** 367-84

[9] G. Vlad, F. Zonca, and S. Briguglio. Dynamics of Alfvén waves in tokamaks. *Rivista del Nuovo Cimento*, 22(7):1–97, 1999.

[10] J.P. Freidberg, *Ideal Magnetohydrodynamics*, Plenum Press, New York, 1987.

[11] H.R. Strauss, *Phys. Fluids*, **20**, (1977) 1354.

[12] R. Izzo, D.A. Monticello, W. Park, J. Manickam, H.R. Strauss, R. Grimm, and K. McGuire, *Phys. Fluids*, **26**, (1983) 2240.

[13] W. Park, S. Parker, H. Biglari, M. Chance, L. Chen, C.Z. Cheng, T.S. Hahm, W.W. Lee, R. Kulsrud, D. Monticello, L. Sugiyama, and R.B. White, *Phys. Fluids B*, **4**, (1992) 2033.

[14] S. Briguglio, G. Vlad, F. Zonca, and C. Kar. Hybrid magnetohydrodynamic-gyrokinetic simulation of toroidal Alfvén modes. *Physics of Plasmas*, 2:3711–3723, 1995.

[15] G. Vlad, C. Kar, F. Zonca, and F. Romanelli. Linear and non-linear dynamics of Alfvén eigenmodes in tokamaks. *Physics of Plasmas*, 2:418–441, 1995. (Also RT/ERG/FUS/94/16 Ass. EUTATOM-ENEA sulla fusione, C.R.E. Frascati (Rome) Italy, 1994).

[16] G. Vlad, S. Briguglio, C. Kar, F. Zonca, and F. Romanelli, in *Theory of Fusion Plasmas*, edited by E. Sindoni and J. Vaclavik, pp. 361–382, Bologna, 1992, Editrice Compositori Società Italiana di Fisica.

[17] A. Bondeson, *Nucl. Fusion*, **26**, (1986) 929.

[18] W.W. Lee, *Phys. Fluids*, **26**, (1983) 556.

[19] W.W. Lee, *J. Comput. Phys.*, **72**, (1987) 243.

[20] T.S. Hahm, *Phys. Fluids*, **31**, (1988) 2670.

[21] T.S. Hahm, W.W. Lee, and A. Brizard, *Phys. Fluids*, **31**, (1988) 1940.

[22] A.M. Dimits, L.L. LoDestro, and D.H.E. Dubin, *Phys. Fluids B*, **4**, (1992) 274.

[23] T. Tajima and F.W. Perkins, private communication, 1983.

[24] T. Tajima, *Computational Plasma Physics*, Addison-Wesley, Redwood City, California, 1989.

[25] M. Kotschenruether, in *Bull. Am. Phys. Soc.*, volume 33, p. 2107, 1988.

[26] S. Briguglio, G. Betello, G. Fogaccia, G. Graziadei, S.C. Guo, C. Kar, F. Romanelli, G. Vlad, and F. Zonca, in *Plasma Physics and Controlled Nuclear Fusion Research 1992, Würzburg*, volume II CN-56/D-2-3, pp. 87–95, Vienna, 1993, International Atomic Energy Agency.

[27] S.E. Parker and W.W. Lee, *Phys. Fluids B*, **5**, (1993) 77.