FAR3d code

User's Guide

Main developers:

D. Spong[1] (spong@ornl.gov)

L. Garcia[2] (lgarcia@fis.uc3m.es)

J. Varela[1] (rodriguezjv@ornl.gov)

[1] Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831-8071

[2] Universidad Carlos III de Madrid, 28911 Leganes, Madrid, Spain

Terms & Conditions of Use

Far3d code is a freely distributed code under the GNU license. The developers will be grateful with the FAR3d users that include references and acknowledge the effort of setting up the code. The users are free to modify the code and to share with the Far3d community new code add-ons, that can be included in future Far3d releases.

Index

0. Quick Start

0.1 Downloading and unpacking the code

Far3d can be downloaded from https://github.com//ORNL-Fusion/FAR3d. Next, extract the files using the command:

> tar xvf FAR3d.tar

The folder FAR3d is created with the next distribution:

FAR3d → /Source : code .f90 files

→ /Input_basic : default values of the input files

→ /Models : code input examples

→ /Help : help files

→ /BOOZ_XFORM : code equilibria input

→ /Addon : external accessory programs

→ User_guide.pdf : Far3d User's Guide

→ Config.sh : compiles the code and creates the user's models

0.2 Run a test case

Several test models are included with FAR3d distribution. The test cases are included in /Models folder, for example the test case "DIIID":

> cd Models/DIIID/

Inside the folder you will find the next files:

a) Input files:

Input_Model

b) Model equilibria:

Eq_DIIID_RS

c) Experimental profiles:

Data.txt

First the user should compile the code executing the bash file Config.sh (see chapter 4) and copy the executable **xfar3d** in the model folder. To run the model, execute the program:

> ./xfar3d

After the end of the run you will find in the same folder the eigenfunctions data and other output files as "farprt" file where you can find the growth rate and frequency of the instability, in this case a Toroidal Alfven Eigenmode.

0.3 Prepare your own model

You can create your own modes using the bash file "Config.sh". First, execute the bash file 'Config.sh':

> ./Config.sh

You need to introduce the name of your new model:

> Welcome to FAR3d, introduce your model name:

> 'Your model name'

> Your model was created at: cd FAR3d/Models/'Your model name'

Now, in the new folder you will find the code executable and the default input files. Modify the input file to adequate the run to your case (a description of the input files is done in the section 2). You will also need the equilibria of the model (in section 6 we explain how to create your model equilibria from a VMEC input using the program 'BOOZ_XFORM' included in Far3d distribution). In addition, you can add experimental profiles if required (see section 2 for further explanations).

1. Code introduction

1.1 FAR3d

The aim of FAR3d code is to offer a fast tool to study the linear MHD and AE stability of nuclear fusion devices. FAR3d code solves the linear reduced resistive MHD equations [1,2,3] adding the Landau damping/growth (wave-particle resonance effects) [4,5] and geodesic acoustic waves (parallel momentum response of the thermal plasma) [6]. FAR3d includes the moment equations of the energetic ion density and parallel velocity allowing treatment of linear wave-ion resonances.

The model requires Landau closure relations that can be calibrated by more complete kinetic models [6]. The simulations are based on an equilibria calculated by the VMEC code [7], but future version will include other equilibria options as SIESTA [8] or EFIT [9].

From the full MHD equations we can derive a reduced set of three equations that follow the evolution of the pressure 'p', the stream function proportional to the electrostatic potential 'Φ' and the perturbation of the poloidal flux 'ψ' retaining the toroidal angle variation 'ζ' (exact three-dimensional equilibrium [1]). The equations for the background or thermal plasma evolution are complemented by moments of the energetic ion equations to add the effect of the wave-particle interaction [10], namely the energetic particle density 'n_f' and velocity moments parallel to the magnetic field lines '$v_{||f}$'. The coefficients of the closure relation are selected to match a two-pole approximation of the plasma dispersion function.

The reduced equations are based on the following assumptions: high aspect ratio, medium thermal ß (of the order of the inverse aspect ratio ε=a/R_0, with 'a' the device minor radius and 'R_0' the major radius), small variation of the fields and small resistivity. The magnetic field and plasma velocity perturbations are defined as:

$$\vec{v} = \sqrt{g} R_0 \vec{\nabla}\zeta \wedge \vec{\nabla}\Phi \qquad ; \qquad \vec{B} = R_0 \vec{\nabla}\zeta \wedge \vec{\nabla}\psi$$

with $\sqrt{g}$ the Jacobian of the coordinate transformation.

1.2 Numerical model

The equations, in dimensionless form, are:

MHD equations:

$$[1]\ \frac{\partial \psi}{\partial t}=\frac{\partial \Phi}{\partial \zeta}+\iota\frac{\partial \Phi}{\partial \theta}+\frac{\eta}{S}J_{\zeta}+\frac{B_0\rho_i^2}{(J-\iota I)}\left(\frac{v_A^2}{v_{T,i}}\right)\left(\frac{\pi}{2}\right)^{1/2}\left|\nabla_{\parallel}\right|Q-\frac{\Lambda\beta_0}{2\varepsilon^2\omega_{cy}}\frac{B_0}{n_0(J-\iota I)}\left(\frac{\partial p}{\partial \zeta}+\iota\frac{\partial p}{\partial \theta}\right)$$

$$[2]\ \frac{\partial U}{\partial t}=-v_{\zeta,eq}\frac{\partial U}{\partial \zeta}-\frac{v_{\theta,eq}}{\rho}\frac{\partial U}{\partial \theta}+\frac{\beta_0}{2\varepsilon^2\rho}\left(\frac{\partial\sqrt{g}}{\partial\theta}\frac{\partial p}{\partial\rho}-\frac{\partial\sqrt{g}}{\partial\rho}\frac{\partial p}{\partial\theta}\right)+\frac{\partial J^{\zeta}}{\partial\zeta}+\iota\frac{\partial J^{\zeta}}{\partial\theta}$$
$$+\frac{\beta_f}{2\varepsilon^2\rho}\left(\frac{\partial\sqrt{g}}{\partial\theta}\frac{\partial n_f}{\partial\rho}-\frac{\partial\sqrt{g}}{\partial\rho}\frac{\partial n_f}{\partial\theta}\right)+\frac{\beta_\alpha}{2\varepsilon^2\rho}\left(\frac{\partial\sqrt{g}}{\partial\theta}\frac{\partial n_\alpha}{\partial\rho}-\frac{\partial\sqrt{g}}{\partial\rho}\frac{\partial n_\alpha}{\partial\theta}\right)$$
$$+\omega_r\rho_i^2\nabla_{\perp}^2U-\frac{\beta_0v_{th,f}^4}{\omega_{cy}^2\omega_r}\mathrm{Im}\left[X_e^{//}+X_i^{//}-\frac{\left(X_i^{/}-X_e^{/}\right)}{2+X_e+X_i}-\upsilon_i(X_i^{/})^2-\upsilon_e(X_e^{/})^2\right]\Omega_d^2(\Phi)$$
$$-\frac{\beta_0(1-\Lambda)}{2\omega_{cy}\varepsilon^2n_0}\left\{\vec{\nabla}\wedge\left[\sqrt{g}\left(\left(\vec{B}\wedge\vec{\nabla}p\right)\cdot\vec{\nabla}\right)\vec{v}_{\perp}\right]\right\}^{\zeta}$$

$$[3]\ \frac{\partial p}{\partial t}=-v_{\zeta,eq}\frac{\partial p}{\partial\zeta}-\frac{v_{\theta,eq}}{\rho}\frac{\partial p}{\partial\theta}-\frac{1}{\rho}\frac{dp_{eq}}{d\rho}\frac{1}{J-\iota I}\left(I\frac{\partial\Phi}{\partial\zeta}-J\frac{\partial\Phi}{\partial\theta}\right)-\Gamma p_{eq}\vec{\nabla}\left(\frac{\vec{B}_0\wedge\vec{\nabla}\Phi}{B_0^2}\right)$$
$$-\frac{\Gamma p_0B_0}{J-\iota I}\left(\tau\frac{\partial}{\partial\theta}+\frac{\partial}{\partial\zeta}\right)v_{\parallel,th}-\frac{\Gamma p_0B_0\sqrt{g}}{J-\iota I}\left[\tau\frac{\partial}{\partial\theta}\left(\frac{1}{\sqrt{g}}\right)+\frac{\partial}{\partial\zeta}\left(\frac{1}{\sqrt{g}}\right)\right]v_{\parallel,th}$$
$$-\frac{\beta_0(1-\Lambda)}{2\omega_{cy}\varepsilon^2n_0\sqrt{g}}p\vec{\nabla}p\cdot\left(\vec{\nabla}\wedge\vec{B}\right)$$

$$[4]\ \frac{\partial v_{\parallel,th}}{\partial t}=-v_{\zeta,eq}\frac{\partial v_{\parallel,th}}{\partial\zeta}-\frac{v_{\theta,eq}}{\rho}\frac{\partial v_{\parallel,th}}{\partial\theta}-\frac{\beta_0B_0}{2n_0(J-\iota I)}\left(\frac{\partial}{\partial\zeta}+\iota\frac{\partial}{\partial\theta}\right)p+\frac{1}{\rho}\frac{\partial\psi}{\partial\theta}\frac{dp_{eq}}{d\rho}$$

Energetic Particle equations:

$$[5]\ \frac{\partial n_f}{\partial t}=-v_{\zeta,eq}\frac{\partial n_f}{\partial\zeta}-\frac{v_{\theta,eq}}{\rho}\frac{\partial n_f}{\partial\theta}-\frac{v_{th,f}^2}{\varepsilon^2\omega_{cy}}\left(\Omega_{dr}\frac{\partial n_f}{\partial\rho}+\Omega_{d\theta}\frac{1}{\rho}\frac{\partial n_f}{\partial\theta}+\Omega_{d\zeta}\frac{\partial n_f}{\partial\zeta}\right)-\frac{n_{f0}B_0}{J-\iota I}\left(\frac{\partial}{\partial\zeta}+\iota\frac{\partial}{\partial\theta}\right)v_{\parallel f}$$
$$-n_{f0}\left(\Omega_{dr}\frac{\partial\Phi}{\partial\rho}+\Omega_{d\theta}\frac{1}{\rho}\frac{\partial\Phi}{\partial\theta}+\Omega_{d\zeta}\frac{\partial\Phi}{\partial\zeta}\right)+n_{f0}\Omega_*(\Phi)$$
$$+\left[\frac{\varepsilon^2\Omega_{cy}}{v_{th,f}^2}\omega_rn_{0f}-\frac{1}{\rho(J-\iota I)}\frac{dn_{0f}}{d\rho}\left(I\frac{\partial}{\partial\zeta}-J\frac{\partial}{\partial\theta}\right)\right]W_{NBI}$$

$$[6]\ \frac{\partial v_{\|f}}{\partial t}=-v_{\zeta,eq}\frac{\partial v_{\|f}}{\partial \zeta}-\frac{v_{\theta,eq}}{\rho}\frac{\partial v_{\|f}}{\partial \theta}+\frac{v_{th,f}^2}{\varepsilon^2\omega_{cy}}\left(\Omega_{dr}\frac{\partial v_{\|f}}{\partial \rho}+\Omega_{d\theta}\frac{1}{\rho}\frac{\partial v_{\|f}}{\partial \theta}+\Omega_{d\zeta}\frac{\partial v_{\|f}}{\partial \zeta}\right)-\sqrt{\frac{\pi}{2}}v_{th,f}\frac{B_0}{J-\imath I}\left|\frac{\partial}{\partial \zeta}+\imath\frac{\partial}{\partial \theta}\right|v$$
$$-\frac{2v_{th,f}^2}{n_{f0}}\frac{B_0}{J-\imath I}\left(\frac{\partial}{\partial \zeta}+\imath\frac{\partial}{\partial \theta}\right)n_f+\frac{v_{th,f}^2}{\rho(J-\imath I)}\frac{1}{n_{0f}}\frac{dn_{0f}}{d\rho}\left(IX_{1,NBI}-JX_{2,NBI}\right)$$

2nd species energetic Particle equations:

$$[7]\ \frac{\partial n_\alpha}{\partial t}=-v_{\zeta,eq}\frac{\partial n_\alpha}{\partial \zeta}-\frac{v_{\theta,eq}}{\rho}\frac{\partial n_\alpha}{\partial \theta}-\frac{v_{th,\alpha}^2}{\varepsilon^2\omega_{cy}}\left(\Omega_{dr}\frac{\partial n_\alpha}{\partial \rho}+\Omega_{d\theta}\frac{1}{\rho}\frac{\partial n_\alpha}{\partial \theta}+\Omega_{d\zeta}\frac{\partial n_\alpha}{\partial \zeta}\right)-\frac{n_{\alpha0}B_0}{J-\imath I}\left(\frac{\partial}{\partial \zeta}+\imath\frac{\partial}{\partial \theta}\right)v_{\|\alpha}$$
$$-n_{\alpha0}\left(\Omega_{dr}\frac{\partial \Phi}{\partial \rho}+\Omega_{d\theta}\frac{1}{\rho}\frac{\partial \Phi}{\partial \theta}+\Omega_{d\zeta}\frac{\partial \Phi}{\partial \zeta}\right)+n_{\alpha0}\Omega_*(\Phi)$$
$$+\left[\frac{\varepsilon^2\Omega_{cy}}{v_{th,\alpha}^2}\omega_r n_{0\alpha}-\frac{1}{\rho(J-\imath I)}\frac{dn_{0\alpha}}{d\rho}\left(I\frac{\partial}{\partial \zeta}-J\frac{\partial}{\partial \theta}\right)\right]W_\alpha$$

$$[8]\ \frac{\partial v_{\|\alpha}}{\partial t}=-v_{\zeta,eq}\frac{\partial v_{\|\alpha}}{\partial \zeta}-\frac{v_{\theta,eq}}{\rho}\frac{\partial v_{\|\alpha}}{\partial \theta}+\frac{v_{th,\alpha}^2}{\varepsilon^2\omega_{cy}}\left(\Omega_{dr}\frac{\partial v_{\|\alpha}}{\partial \rho}+\Omega_{d\theta}\frac{1}{\rho}\frac{\partial v_{\|\alpha}}{\partial \theta}+\Omega_{d\zeta}\frac{\partial v_{\|\alpha}}{\partial \zeta}\right)-\sqrt{\frac{\pi}{2}}v_{th,\alpha}\frac{B_0}{J-\imath I}\left|\frac{\partial}{\partial \zeta}+\imath\frac{\partial}{\partial \theta}\right|v$$
$$-\frac{2v_{th,\alpha}^2}{n_{\alpha0}}\frac{B_0}{J-\imath I}\left(\frac{\partial}{\partial \zeta}+\imath\frac{\partial}{\partial \theta}\right)n_\alpha+\frac{v_{th,\alpha}^2}{\rho(J-\imath I)}\frac{1}{n_{0\alpha}}\frac{dn_{0\alpha}}{d\rho}\left(IX_{1,\alpha}-JX_{2,\alpha}\right)$$

Auxiliary equations:

$$[9]\quad 0=Q-\nabla_\perp^2\psi$$

$$[10]\ 0=\left(1+\rho_f^2\nabla_\perp^2\right)W_{NBI}-\rho_f^2\nabla_\perp^2\Phi \quad [11]\ 0=\left(1+\rho_f^2\nabla_\perp^2\right)X_{1,NBI}-\frac{\partial\psi}{\partial\zeta} \quad [12]\ 0=\left(1+\rho_f^2\nabla_\perp^2\right)X_{2,NBI}-\frac{\partial\psi}{\partial\theta}$$
$$[13]\ 0=\left(1+\rho_f^2\nabla_\perp^2\right)W_\alpha-\rho_f^2\nabla_\perp^2\Phi \quad [14]\ 0=\left(1+\rho_f^2\nabla_\perp^2\right)X_{1,\alpha}-\frac{\partial\psi}{\partial\zeta} \quad [15]\ 0=\left(1+\rho_f^2\nabla_\perp^2\right)X_{2,\alpha}-\frac{\partial\psi}{\partial\theta}$$

Here, $U=\sqrt{g}\left[\vec{\nabla}\wedge\left(n_0\sqrt{g}v\right)\right]^\zeta$ is the vorticity and n_0 the ion and electron mass density. The toroidal current density J^ζ is defined as:

$$J^\zeta=\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(-\frac{g_{\rho\theta}}{\sqrt{g}}\frac{\partial\psi}{\partial\theta}+\rho\frac{g_{\theta\theta}}{\sqrt{g}}\frac{\partial\psi}{\partial\rho}\right)-\frac{1}{\rho}\frac{\partial}{\partial\theta}\left(-\frac{g_{\rho\rho}}{\sqrt{g}}\frac{1}{\rho}\frac{\partial\psi}{\partial\theta}+\rho\frac{g_{\rho\theta}}{\sqrt{g}}\frac{\partial\psi}{\partial\rho}\right)$$

$v_{\zeta,eq}$ is the equilibrium toroidal rotation, $v_{\theta,eq}$ is the equilibrium poloidal rotation and $v_{\|,th}$ is the parallel velocity of the thermal particles. n_f and n_α are the density of the energetic particles species normalized to the density at the magnetic axis, Φ to a^2B_0/τ_{A0} and ψ to a^2B_0, with $\tau_{A0}=R_0(\mu_0 n_0)^{1/2}/B_0$ the Alfven

time. All lengths are normalized to a generalized minor radius, the time to the resistive time, the magnetic field to B_0 (the averaged value at the magnetic axis) and the pressure to its equilibrium value at the magnetic axis. The Lundquist number S is the ratio of the resistive time to the Alfven time. ι is the rotational transform, $v_{th,f} = \sqrt{T_f / m_f}$ the energetic particle thermal velocity normalized to the Alfven velocity in the magnetic axis and ω_{cy} the energetic particle cyclotron frequency normalized to the Alfven time. q_f is the charge, T_f the temperature and m_f the mass of the energetic particles. I is the electric current and J the current density

The Ω_{dr} operator is constructed to model the average drift velocity of a passing particle:

$$\Omega_d = -\frac{v_{th,f}^2 v_A}{\omega_{cy} R_0}\left(\Omega_{dr}\frac{\partial}{\partial\rho} + \Omega_{d\theta}\frac{\partial}{\partial\theta} + \Omega_{d\zeta}\frac{\partial}{\partial\zeta}\right)$$

where:

$$\Omega_{dr} = \frac{\sqrt{g}}{2\rho\varepsilon^2(J+\iota I)}\left(I\frac{\partial}{\partial\zeta}\frac{1}{\sqrt{g}} - \frac{\partial}{\partial\theta}\frac{1}{\sqrt{g}}\right)$$

$$\Omega_{d\theta} = \frac{\sqrt{g}}{2\varepsilon^2(J+\iota I)^2}\left[J\frac{\partial}{\partial\rho}(J+\iota I)\frac{1}{\sqrt{g}} - \beta_*\rho(J+\iota I)\frac{\partial}{\partial\zeta}\frac{1}{\sqrt{g}}\right]$$

$$\Omega_{d\zeta} = \frac{\sqrt{g}}{2\rho\varepsilon^2(J+\iota I)^2}\left[\beta_*\rho(J+\iota I)\frac{\partial}{\partial\theta}\frac{1}{\sqrt{g}} - I\frac{\partial}{\partial\rho}(J+\iota I)\frac{1}{\sqrt{g}}\right]$$

The Ω_* operator is constructed to model the diamagnetic drift frequency:

$$\Omega_* = \frac{T_f}{q_f B_0 a^2 \rho (J+\iota I)}\frac{1}{n_{0f}}\frac{dn_{0f}}{d\rho}\left(I\frac{\partial}{\partial\zeta} - J\frac{\partial}{\partial\theta}\right)$$

Equations [3] and [4] introduce the parallel momentum response of the thermal plasma, required for coupling to the geodesic acoustic waves accounting the geodesic compressibility in the frequency range of the geodesic acoustic mode. Equations [1] and [2] include the effect of the ion finite Larmor radius effect (third and fifth terms respectively), with ρ_i the thermal ions Larmor radius and $v_{T,i}$ the ion thermal velocity. Equation [2] introduces the effect of the electron-ion Landau damping and collisions (sixth term) with ω_r the frequency of the instability. In the equations [5] and [6] the effect of the energetic particle finite Larmour radius is added by the auxiliary equations [10 – 12] (last term).

Equations [1] to [3] include the two fluid terms (last terms) with Λ the electron pressure / total pressure ratio.

We use the Boozer formulation for the flux coordinates [11] where the Jacobian of the coordinate transformation is:

$$\sqrt{g} = \frac{J - \iota I}{B^2}$$

The equations are written using the equilibrium flux coordinates (ρ, θ, ζ) with ρ the generalized radial coordinate proportional to the square root of the toroidal flux function (normalized to one at the edge) and θ the poloidal angle.

The FAR3D code uses finite differences in the radial direction and Fourier expansions in the two angular variables. The numerical scheme is semi-implicit in the linear terms. The nonlinear version uses a two semi-step method to ensure Δt^2 accuracy. The functions have two components, equilibrium and perturbation, represented as $A = A_{eq} + \hat{A}$.

Present Far3d version is serial although future code version already in test phase will implement MPI and OPENMP capabilities.

For more information about the numerical model and code implementation please see the pdf files included in the folder /Help. You will find the next documents: Acoustic_modes.pdf, Boozer_coordinates.pdf, FAR3d_basic.pdf, FLR_effects.pdf, Gyrofluid_model.pdf, MHD_model_implementation.pdf, Toroidal_rotation.pdf, MHD_model.pdf, Two_fluid.pdf and Landau_elec_ion_damp.pdf.

The model was validated on previous studies of AE activity in LHD [12,13,14,15,16], TJ-II [17], W-7X [18], DIII-D [19] and ITER [20] indicating a reasonable agreement with the experimental observations.

2. Input files

Far3d input file is:

Input_Model

!!!! We suggest to the users don't modify the structure of the input files to avoid the malfunction of the code. If you need to modify the input files, you should also modify the subroutine '*inputlist*'

2.1 Input file: **Input_Model**

In the first part of the input file the user can configure the simulation indicating the number of grid point, modes and number of time steps between other main parameters. In addition, you can activate the different code modules, for example the finite Larmor radius effects or introducing experimental profiles. The list of available modules and simulation options are the following:

nstres: indicates if the run is a new run if 0 new run or a continuation if 1.

numrun: (4 digit number) run number.

numruno: (4 digits number + z) name of the previous run output.

numvac: (5 digit number) index of the run number.

nonlin: linear run if 0, non linear run if 1 (no available yet).

ngeneq: indicates the equilibrium input of the model (only VMEC available now).

eq_name: equilibrium name.

maxstp: number of simulation time steps.

dt0: simulation time step.

ldim: total number of poloidal modes (equilibrium + dynamic).

leqdim: equilibrium poloidal modes.

jdim: number of radial points.

ext_prof: include experimental profiles if 1. If 0, the user must include the profiles in the file **Input_Model_Prof.**

ext_prof_name: name of the external profile file.

ext_prof_len: number of lines in the external profile.

iflr_on: index that activates the thermal ion FLR damping effects if 1.

epflr_on: index that activates the fast particle FLR damping effects if 1.

ieldamp_on: index that activates electron-ion Landau damping effect if 1 (this only is only available if **ext_prof=1**).

twofl_on: index that activates the two fluid effects if 1.

alpha_on: index that activates a 2nd fast particle species if 1.

Trapped_on: index that activates helically trapped 1st fast particle species if 1.

matrix_out: index to generate the eigensolver input files.

M0dy: index to indicate the number of equilibrium modes that evolves in time (only in nonlinear simulations, not available yet).

In the second part of the input file the user can configure the simulation parameters, introducing the list of dynamic and equilibrium modes, energetic particle ß, diffusivities, Landau closure terms between others. The list of available model parameters are the following:

mm: list of poloidal dynamic and equilibrium modes.

nn: list of toroidal dynamic and equilibrium modes.

mmeq: list of poloidal equilibrium modes.

nneq: list of toroidal equilibrium modes.

!!!!! ATTENTION: **ldim = total number of mm and leqdim = total number of mmeq !!!!!**

ipert: different options to drive a perturbation in the equilibria. There are two options:

If ipert=0 the mode 0/0 is not perturbed.

If ipert=1 the mode 0/0 is also perturbed.

widthi: size of the perturbation.

Auto_grid_on: the grid spacing is automatic selected if 1 with:

ni=(jdim/2) – 1 ; nis=(jdim/4) + 1 ; ne=jdim/4

ni: number of points interior to the island.

nis: number of points in the island.

ne: number of points exterior to the island.

!!!!! ATTENTION: **jdim = ni + nis + ne if Auto_grid_on = 0 !!!!!**

delta: normalized width of the uniform fine grid (island).

rc: center of the fine grid (island) along the normalized minor radius.

gamma: adiabatic index.

s: magnetic Lundquist number.

betath_factor: this factor multiplies the thermal beta to reduce or increase its value in the simulation. It should be noted that, if the factor is large, the flux surfaces calculated by VMEC are not valid and the equilibria must be recalculated.

ietaeq: index to select the resistivity profile. There are three options:

If ietaeq=1 the resisitivity is calculated using the electron temperature.

If ietaeq=2 no radial dependency. The user should include the parameter '**etascl**'.

If ietaeq=3 the resistivity is defined by the profile $\eta = eta0\left(1+\left(\frac{r}{reta}\right)^{2etalmb}\right)^{1/etalmb}$

where '**reta**' and '**etalmb**' must be introduced by the user.

bet0_f: fast particle beta.

bet0_falp: 2nd species fast particle beta.

omcy: normalized fast particle cyclotron frequency.

omcyb: normalized bounce frequency of the helically trapped fast particles.

rbound: normalized bounce distance of the helically trapped fast particle.

omcyalp: normalized 2nd species fast particle cyclotron frequency

itime: time normalization option.

dpres: electron pressure normalized to the total pressure (two fluid effects only).

stdifp: thermal pressure eq. diffusivity.

stdifu: vorticity eq. diffusivity.

stdifv: thermal plasma parallel velocity eq. diffusivity.

stdifnf: fast particle density eq. diffusivity.

stdifvf: fast particle parallel velocity eq. diffusivity.

stdifnfalp: fast particle density eq. diffusivity.

stdifvfalp: fast particle parallel velocity eq. diffusivity.

LcA0: Landau closure 1.

LcA1: Landau closure 2.

LcA2: correction to the fast particle beta.

LcA3: correction to the ratio between fast particle thermal velocity and Alfven velocity.

LcA0alp: Landau closure 1 2nd species.

LcA1alp: Landau closure 2 2nd species.

LcA2alp: correction to the 2nd species fast particle beta.

LcA3alp: correction to the ratio between fast particle thermal velocity and Alfven velocity 2nd species.

omegar: eigenmode frequency without damping effects.

iflr: thermal ions larmor radius normalized to the minor radius.

r_epflr: energetic particle larmor radius normalized to the minor radius.

r_epflralp: 2nd species energetic particle larmor radius normalized to the minor radius.

lplots: number of eigenfunction modes included in the output files.

nprint: number of step for an output in farprt file.

ndump: number of step for an output.

DIIID_u: turn on to use the same units than TRANSP output in the external profiles (so the code assumes the input is in cm, not m).

In the third part of the input file the user can introduce in the model user defined profiles, for example the energetic particle density profile, thermal electron temperature or displace the iota profile. The list of available user defined profiles are the following:

EP_dens_on: this index activates user defined fast particle density profile if 1, defined as $n_f(j) = \frac{0.5 \cdot (1 + tanh(Adens \cdot (Bdens - r(j)))) + 0.02}{0.5 \cdot (1 + tanh(Adens \cdot Bdens)) + 0.02}$.

Adens: fast particle density profile flatness.

Bdens: location of the fast particle density profile gradient.

Alpha_dens_on: this index activates user defined 2nd species fast particle density profile if 1, defined as $n_\alpha(j) = \frac{0.5 \cdot (1 + tanh(Adensalp \cdot (Bdensalp - r(j)))) + 0.02}{0.5 \cdot (1 + tanh(Adensalp \cdot Bdensalp)) + 0.02}$.

Adensalp: 2nd species fast particle density profile flatness

Bdensalp: location of the 2nd species fast particle density profile gradient

EP_vel_on: this index activates the used defined 1st species fast particle of $v_{th,f} / v_{A0}$ profile if 1.

Alpha_vel_on: this index activates the user defined 2nd species fast particle $v_{th,\alpha} / v_{A0}$ profile if 1.

q_prof_on: the safety factor profile from external profiles if 1.

Eq_vel_on: the thermal plasma eq. toroidal velocity from external profiles if 1.

Eq_velp_on: this index activates the external profile of the equilibrium poloidal rotation profile if 1.

Eq_Presseq_on: this index activates the external profile of the equilibrium thermal pressure profile if 1.

Eq_Presstot_on: this index activates the external profile of the equilibrium thermal + EP pressure profiles if 1.

deltaq: this parameter introduces a displacement in the safety factor profile (only tokamak eq.).

deltaiota: this parameter introduces a displacement in the iota profile (only stellarator eq.).

etascl: flat resistivity value.

eta0: user defined resistivity profile parameter.

reta: user defined resistivity profile parameter.

etambl: user defined resistivity profile parameter.

cnep: user defined thermal plasma density profile. 11 index should be included to define the profile $n_0(j) = cnep(0) * r^{2 \cdot cnep(i)}(j)$, with i=1,10.

ctep: user defined thermal electron plasma temperature profile. 11 index should be included to define the profile $T_e(j) = ctep(0) * r^{2 \cdot ctep(i)}(j)$, with i=1,10.

cnfp: user defined energetic particles density profile. 11 index should be included to define the profile $n_f(j) = cnfp(0) * r^{2 \cdot cnfp(i)}(j)$, with i=1,10.

cvep: user defined thermal ions parallel velocity profile (only for thermal ion FLR effects). 11 index should be included to define the profile $T_i(j) = cvep(0) * r^{2 \cdot cvep(i)}(j)$, with i=1,10.

cvfp: user defined energetic particles parallel velocity profile. 11 index should be included to define the profile $v_{th,f} / v_{A0}(j) = cvfp(0) * r^{2 \cdot cvfp(i)}(j)$, with i=1,10.

cnfpalp: user defined 2nd species energetic particles density profile. 11 index should be included to define the profile $n_\alpha(j) = cnfpalp(0) * r^{2 \cdot cnfpalp(i)}(j)$, with i=1,10.

cvfpalp: user defined 2nd species energetic particles parallel velocity profile. 11 index should be included to define the profile $v_{th,\alpha} / v_{A0}(j) = cvfpalp(0) * r^{2 \cdot cvfpalp(i)}(j)$, with i=1,10.

eqvt: user defined thermal plasma equilibrium toroidal velocity. 11 index should be included to define the profile $v_{\zeta,eq} / v_{A0}(j) = eqvt(0) * r^{2 \cdot eqvt(i)}(j)$, with i=1,10.

eqvp: user defined thermal plasma equilibrium poloidal velocity. 11 index should be included to define the profile $v_{\zeta,eq} / v_{A0}(j) = eqvt(0) * r^{2 \cdot eqvt(i)}(j)$, with i=1,10.

2.2 Continue a simulation:

To continue a simulation you should modify three elements in the input file **Input_Model.** First **nstres = 1**, to indicate that the run is a continuation. Second you should introduce the index name (**numrun**) of the continuation run and indicate the output restart file name of the previous run (**numruno**), that has the format '*fsXXXXz*' with *XXXX* the **numrun** index of the previous run. For example, if **numrun = 0000** is the run index of the previous run, the output restart file name is **fs0000z**, so to continue this run we introduce a new run index, for example **numrun = 0001** and we the name of the output restart file name **numruno = 0000z.**

3. Experimental profiles

The input file **Input_Model** includes the option to activate experimental profiles in the simulation if **ext_prof=1.** The user must introduce the name of the file **ext_prof_name** and the length of the data rows **ext_prof_len**:

First, there is a file **header** that includes several plasma parameters:

PLASMA GEOMETRY

Vacuum Toroidal magnetic field at R=1.69550m [Tesla]

2.036657

Geometric Center Major radius [m]

1.752689

Minor radius [m]

0.602

Avg. Elongation

1.80556

Avg. Top/Bottom Triangularity

0.490581

Main Contaminant Species

12C

Main Ion Species mass/proton mass

2.0

TRYING TO GET TO BETA(0)=0.05666 , Rmax=1.752689

Next, a list of columns with the experimental data (from left to right):

Rho: the normalized square value of the toroidal flux

q: safety factor

Beam Ion Density: the density of the energetic particles in $10^{20}m^{-3}$ units.

Ion Density: the thermal ion density in $10^{20}m^{-3}$ units.

Elec Density: the thermal electron density in $10^{20}m^{-3}$ units.

Impurity Density: the impurity particles density in $10^{20}m^{-3}$ units.

Beam Ion Effective: the temperature of the energetic particles in keV units.

Ion Temp: the temperature of the thermal ions in keV units.

Electron Temp: the temperature of the thermal electrons in keV units.

Beam Pressure1: the pressure of the energetic particles in kPa units.

Thermal Pressure: the pressure of the thermal particles in kPa units.

Equil.Pressure: the equilibrium pressure in kPa units.

Tor Rot: the plasma toroidal rotation in km/s units.

Pol Rot: the plasma poloidal rotation in km/s units.

If the user activates the module of the 2nd energetic particle species **alpha_on=1** with experimental profiles, the file needs to have two extra columns, one after the **Elec Density** column and another after the **Electron Temp** column defined as:

Alpha Density: the density of the 2nd energetic particle species in $10^{20}m^{-3}$ units.

Alpha Temp: the temperature of the 2nd energetic particle species in keV units.

There is an experimental profile option in the input file **Input_Model_Param** called **DIIID_u**. If **DIIID_u=1** the user can use directly the output of TRANSP code as the experimental profiles input. In this case the densities are in $10^{13}cm^{-3}$ units and there are 2 extra columns after the **Equil.Pressure** column:

Zeff: the atomic number of the impurities

Tor Rot: the plasma toroidal rotation frequency in kHz units.

4. Config.sh

The Config.sh bash file creates the user's model. The structure of the bash file is the following:

```
#-----------------------------------------------------------------------------------------------------

# Compilation of FAR3d executable

#-----------------------------------------------------------------------------------------------------


echo Welcome to FAR3d, introduce your model name:

read varname

cd Models

mkdir $varname

cd ../Source


#These are generic flags and commands.

Comp="gfortran"

Exc=" xfar3d"

Opt=" -O2"

Flag=" -ffree-line-length-none"


echo The FAR3d executable is been compiled, please wait...


#Main program files

OBJS=" Main.f90 Modules.f90"
```

```
#Setup subroutines files

OBJS2=" inputlist.f90 dfault.f90 setup.f90 setmod.f90 seteq.f90 etachi.f90 grid.f90 pert.f90 vmec.f90 ae_profiles.f90"

#Model subroutines

OBJS3=" b2lx.f90 block.f90 linstart.f90 lincheck.f90 linstep.f90"

#Operator subroutines

OBJS4=" clgam.f90 derivatives.f90 eqdy_dyeq.f90 om.f90 dlstar.f90 eigensolver_tools.f90 Laundamp_tools.f90 dlsq.f90"

#Output subroutines

OBJS5=" endrun.f90 energy.f90 output.f90 wrdump.f90 rddump.f90"

#Tool subroutines

OBJS6=" mult.f90 cnvt.f90 decbt.f90 solbt.f90 quadq.f90 functions.f90 mmlims.f90 numinc.f90 eqsplns.f90 elapsed_time.f90"


#Code compilation

$Comp -o $Exc $Opt $Flag $OBJS $OBJS2 $OBJS3 $OBJS4 $OBJS5 $OBJS6


mv xfar3d ../Models/$varname/xfar3d

cd ../Input_basic

cp * ../Models/$varname


echo Your model was created at: cd FAR3d/Models/$varname

#-------------------------------------------------------------------------------------------------------
```

The main elements in the Config.sh are:

varname: name of the model introduced by the use during the bash execution.

Comp: the compiler (gfortran by default).

Exc: the name of the Far3d executable (xfar3d by default).

Opt: Optimization flag (-O2 by default).

Flag: Other compiler flags (-ffree-line-length-none by default).

OBJS: List of f.90 files required to compile the executable divided in groups

If the user wants to change the compiler or flags the Config.sh must be modified accordingly.

After Config.sh bash execution a new model it will created inside the folder /Model with the name given by the user /Model/$ **varname.**

!!!! We suggest to the users don't modify the structure of the Config.sh bash, particularly the list of **OBJS,** because the consequence will be a compiler malfunction. If the users develop a new subroutine, it should be included in the list and properly referenced by the rest of code elements.

5. Code output

The Far3d outputs consist is the following files:

farprt: Main FAR3d output file.

profiles.dat: normalized profiles used in the run.

profiles_ex.dat: experimental profiles used in the run (only if **ext_prof=1**).

egn_eval_1.out: Mode matrix.

br: radial magnetic field eigenfunction.

bth: poloidal magnetic field eigenfunction.

phi: stream function proportional to the electrostatic potential eigenfunction.

pr: pressure eigenfunction.

psi: poloidal flux eigenfunction.

uzt: vorticity eigenfunction.

vr: radial velocity eigenfunction.

vth: poloidal velocity eigenfunction.

vthprlf: thermal parallel velocity eigenfunction.

nf: energetic particle density eigenfunction.

vparf: energetic particle parallel velocity eigenfunction.

nalp: 2nd species energetic particle density eigenfunction.

vparalp: 2nd species energetic particle parallel velocity eigenfunction.

spctr: modes kinetic and magnetic energy.

5.1 Output file: **farprt**

This is the main output file of FAR3d. The file structure is the following:

- Copy of the three input files.

- Copy of several parameters defined in **control**, **domain**, **equil** and **dynamo** modules.

- Mode matrix generated by the code in **setmod** subroutine

- Energy and growth rate of the modes included in the simulation each **nprint** time steps.

- Growth rate and frequency of the instability for each mode. If the code is converged, all modes must have a similar value.

5.2 Output file: **profiles.dat**

This file includes the main simulation profiles normalized to the value in the magnetic axis. The profiles included are, from left to right:

a) If **ext_prof=0**

- **r**: normalized minor radius.

- **denseq**: thermal plasma density.

- **teeq**: thermal electron temperature.

- **nfeq**: energetic particle density.

- **dnfeqdr**: radial derivative of the energetic particle density.

- **vfova**: energetic particle thermal velocity normalized to the Alfven velocity in the magnetic axis.

- **cureq**: electric current.

- **feq**: current density.

- **pressure**: thermal plasma pressure.

- **iota**: iota profile (no normalization).

- **1/D**: inverse of the metric Jacobian for n=0 (no normalization).

- **curvature**: radial derivative of the metric Jacobian forn=0 (no normalization).

- **shear**: magnetic shear (no normalization).

- **eta**: plasma resistivity.

b) If **ext_prof=1**

- **r**: normalized minor radius.

- **dnnbi**: energetic particle density.

- **dne**: thermal plasma electron density.

- **dni**: thermal ion density normalized.

- **dnalpha**: 2nd species energetic particle density (only if **alpha_on=1**)

- **tbn**: energetic particle temperature.

- **ti**: thermal ions temperature.

- **te**: thermal electrons temperature.

- **talpha**: 2nd species energetic particle temperature (only if **alpha_on=1**)

- **vfova**: energetic particle thermal velocity normalized to the Alfven velocity in the magnetic axis.

- **cureq**: electric current.

- **feq**: current density.

- **pressure**: thermal plasma pressure.

- **iota**: iota profile (no normalization).

- **1/D**: inverse of the metric Jacobian for n=0 (no normalization).

- **curvature**: radial derivative of the metric Jacobian forn=0 (no normalization).

- **shear**: magnetic shear (no normalization).

- **eta**: plasma resistivity (no normalization).

5.3 Output file: **profiles_ex.dat**

This file includes the profiles used in the simulation. The profiles included are, from left to right:

- **r**: normalized minor radius.

- **dnnbi**: energetic particle density (m^{-3}).

- **dne**: thermal plasma electron density (m^{-3})..

- **dni**: thermal ion density normalized (m^{-3})..

- **dnalpha**: 2nd species energetic particle density (m^{-3}). Only if **alpha_on=1**.

- **tbn**: energetic particle temperature (eV).

- **ti**: thermal ions temperature (eV).

- **te**: thermal electrons temperature (eV).

- **talpha**: 2nd species energetic particle temperature (eV). Only if **alpha_on=1**.

- **vzt_eq**: equilibrium thermal plasma toroidal velocity (m/s).

- **vth_eq**: equilibrium thermal plasma toroidal velocity (m/s).

- **vthermaleq**: thermal ions = velocity (m/s).

- **eta**: plasma resistivity ($m^3 kg/(s\ C^2)$).

6. Create your model equilibria

FAR3d requires as input a **VMEC** equilibria transformed to Boozer coordinates. The coordinate system transformation is done by the code '**BOOZ_XFORM**', also included in the FAR3d distribution, at the folder with the same name. The structure of /BOOZ_XFORM folder is the following:

/BOOZ_XFORM → /Source : code .f90 files

→ /Equilbria : collection of equilibria input

VMEC code can be obtained from the distribution **STELLOPT**. Once installed **VMEC**, you need the input file **input.(case name)** to create the equilibria, executing the program:

> xvmec2000 input.(case name)

VMEC generates the next output files:

jxbout.(case name).txt

mercier.(case name)

threeed1.(case name)

wout_(case name).nc

The FAR3d input is generated executing **BOOZ_XFORM** using the input file **in_booz.(case_name)**:

> xbooz_xform in_booz.(case_name) far

The program **BOOZ_XFORM** creates the next files:

bmnc_booz.txt, gimnc_booz.txt, gmnc_booz.txt, grrmnc_booz.txt, grrojmnc_booz.txt, grtmnc_booz.txt, grtojmnc_booz.txt, gttmnc_booz.txt, gttojmnc_booz.txt, jbgrrmnc_booz.txt, jbgrtmnc_booz.txt, jbgttmnc_booz.txt, phi_1b.txt, phi_2b.txt, phi_3b.txt, pmns_booz.txt, prfeq_vmec.txt, R_Z_1.txt,

R_Z_1b.txt, R_Z_2.txt, R_Z_2b.txt, R_Z_3.txt, R_Z_3b.txt, rmnc_booz.txt, sum_gmncb.txt, woutb and zmns_booz.txt

The input file of FAR3d is **woutb** file. The rest of outputs are text file with info of the transformation.

The **BOOZ_XFORM** source included in the FAR3d distribution was modified respect to the Stellopt distribution. There are 5 new subroutines: boozer_metric.f, write_polcut.f, root.f, vcoords_gijb.f and boozer_gij.f. Also, the next subroutines are modified: boozer_xform.f, booz_params.f, boozer.f and allocate_boozer.

7. Code Add-ons

FAR3d add-ons are external programs of the main distribution that must be compiled and executed independently. These programs include new analysis tools using FAR3d output.

7.1 Eigensolver.

The eigensolver program Xjdqz is compiled using the script "Eigensover.sh" located at /FAR3d/Addon and requires the library libjdqz.a. The source to build this library can be obtained from:

http://www.staff.science.uu.nl/~sleij101/JD_software/jd.html.

In addition, the packages LAPACK and LBLAS should be installed in the computer.

After running the shell, the executable "xEigen" is created at /Addon/Eigensolver.

Another option is to use the two matrix files with Jacobi-Davidson solvers in the SLEPC library (http://slepc.upv.es). Some limited testing has been done with the Slepc solvers, indicating agreement with results from xjdqz and xae3d solvers. SLEPC requires the installation of PETSC (must be the same versions), and has the advantage over xjdqz that it is parallelized and can be used for very large eigenvalue problems.

To use the eigensolver follow the next steps:

a) Activate the generation of the eigensolver input: **matrix_out** parameter in **Input_Model** (.true.).

b) Execute FAR3d.

c) The eigensolver input is created in the model folder: a_matrix.dat, b_matrix.dat and jdqz.dat.

d) The next message will appear after the FAR3d execution:

```
================================================
=======FAR3d Eigensolver module activated========
================================================
Execute xEigen located at /FAR3d/Addon/Eigensolver
================================================
==./../../xEigen frq grwth=======================
==frq : reference normalized mode frequency======
==grwth : reference normalized growth rate=======
================================================
```

This message indicates that the eigensolver input is created.

e) Execute the eigensolver program from the model folder:

./../../../Addon/Eigensolver/xEigen (normalized frequency) (normalized growth rate).

The eigensolver output are two files:

egn_values.dat: file with two columns, first the normalized mode growth rate and second the normalized mode frequency

egn_mode_asci.dat: file with one column, number of modes calculated, number of poloidal modes /2, number of grid points, list of poloidal number and toroidal numbers, radial grid points, list of Φ eigen-functions.

To create a set of files where the data of the simulation is divided in different mode target where the poloidal modes are ordered in different columns, the program columns.f90 is included in the eigensolver folder. It requires the file **egn_mode_asci.dat**

a) Compile the program columns.f90 and create the executable c_AE.

b) Run c_AE.

X. References

[1] Garcia, L., Proceedings of the 25th EPS International Conference, Prague, 22A, Part II, 1757, (1998)

[2] Charlton, L. A. et al, Journal of Comp. Physics, 63, 107, (1986).

[3] Charlton, L. A. et al, Journal of Comp. Physics, 86, 270, (1990).

[4] Spong, D. A. et al, Phys. Fluids B, 4, 3316, (1992).

[5] Hedrick, C. L. et al, Phys. Fluids B, 4, 3869, (1992).

[6] Spong, D. A. et al, Nucl. Fusion, 53, 053008, (2013).

[7] Hirshman, S. P. et al, Phys. Fluids, 26, 3553, (1983).

[8] Hirshman, S. P. et al, Phys. of Plasmas, 18, 062504, (2011).

[9] Lao L.L., et al, Nucl. Fusion, 25, 1611, (1985).

[10] Hammett, G. W. et al, Phys. Rev. Lett, 64, 3019, (1990).

[11] Boozer, A.H., Phys. Fluids, 25, 520, (1982).

[12] Varela, J. et al, Nucl. Fusion, 57, 046018, (2017).

[13] Varela, J. et al, Phys. Plasma, 19, 082501, (2012).

[14] Varela, J. et al, Phys. Plasma, 19, 082512, (2012).

[15] Varela, J. et al, Phys. Plasma, 21, 032501, (2014).

[16] Varela, J. et al, Phys. Plasma, 21, 092505, (2014).

[17] Varela, J. et al, Nucl. Fusion, 57, 126019, (2017).

[18] Varela, J. et al, *'Alfven Eigenmodes stability in 3D configurations using a Landau-closure model'*, IAEA 17th Energetic Particle meeting, Princeton, P36 (2017).

[19] Varela, J. et al, *'Optimization of DIII-D discharges to avoid AE destabilization'*, APS, Milwaukee, GP11.100 (2017)

[20] Garcia, L. , et al, *'Alfven Eigenmode driven by alpha particles and NBI energetic particles'*, EPS 45th, Prague (2018)