stella:

Technical Documentation and User Manual

Version 0 (in preparation, last revised, December 11, 2018) `stella`is a code ... available at

https://github.com/mabarnes/stella

You are very welcome to use it and/or to contribute to its development. It is also under continuous development, which means that it might contain minor bugs and that its documentation may be incomplete. We are very happy to receive any feedback, which you can send us to

hello@world.com.

`stella` is a numerical code for the the study of gyro-kinetic stability and turbulence in multi-species stellarator plasmas. It counts with the participation of researchers from the the University of Oxford (United Kingdom), University of Marylad (USA) and National Fusion Laboratory at CIEMAT (Spain)...

* The work is carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053.

Contents

Chapter 1

Introduction

1.1 Motivation

1.2 Equations

Chapter 2

Downloading, installing and running `stella`

...

2.1 Downloading from the public repository

2.2 Setting up `stella` on a new system

The first time that you try to run `stella`, you may need to install some software, and to set a few environment variables.

2.2.1 System requirements

In order to compile and run `stella`, your system should already have installed:

- The `PETSc` library for solving large linear systems.
- An implementation of `MPI`for parallel runs.
- `FFTW` for computing the discrete Fourier transform.

Chapter 3

Inputs

3.1 Geometry input: the `wout*.nc` file

3.2 The physics case input: the `*.in` file

`stella` uses for its execution a single input file. In this section, we describe how this is structured in different namelists and which input variables can be included in each of these. The input file, which should include the suffix `.in` at the end of its name, is structured if the following namelists:

- `zgrid_parameters`
- `geo_knobs`
- ...

In the following sections the description of the different input variables that each namelist has is provided. The purpose, description, variable type, etc of each variable is displayed in tabular format, where the header shows the name of the variable in teletype font family with the correponding mathematial mode symbol used in other parts of the documents, if any.

3.2.1 The `zgrid_parameters` namelist

`nzed` (N_ζ)	
Description	Variable for the grid of the spatial grid domain along ζ. It sets the number of divisions of the magnetic field line for the chosen flux tube along the coordinate ζ.
Type	integer
Default	

`nfield_periods`

Description	It defines the flux tube length along ζ. Being $N_{\rm fp}$ the number of field periods per 2π toroidal segment of a given equilibrum (e.g. $N_{\rm fp} = 5$ for W7-X), `nfield_periods` is the number of toroidal field periods the flux tube extents along the ζ direction. Considering the safety factor q at a given flux surface, set through the variable `torflux`, if the flux tube is wanted to extent along N_θ poloidal turns, i.e. covering the range $(-N_\theta\pi, N_\theta\pi)$ in θ, the following rule can be written to set accordingly `nfield_periods`number of divisions of the magnetic field line for the chosen flux tube along the coordinate ζ. $$\texttt{nfield\_periods} = qN_{\rm fp}N_\theta \quad (3.1)$$
Type	float
Default	

3.2.2 The `zgrid_parameters` namelist

3.2.3 The `geo_knobs` namelist

3.2.4 The `vmec_parameters` namelist

3.2.5 The `parameters` namelist

3.2.6 The `vpamu_grids_parameters` namelist

3.2.7 The `dist_fn_knobs` namelist

adiabatic_option

Description	Form of the adiabatic response (if a species is being modeled as adiabatic). Ignored if there are electrons in the species list.
Values	• `no-field-line-average-term`: adiabatic species has $n = \varphi$. Appropriate for single-species ETG simulations. • `field-line-average-term`: adiabatic species has $n = \varphi - \langle\varphi\rangle$. Appropriate for single-species ITG simulations. • `iphi00=0`: same as `no-field-line-average-term`. • `iphi00=1`: same as `no-field-line-average-term`. • `iphi00=2`: same as `field-line-average-term`. • `iphi00=3`: adiabatic species has $n = \varphi - \langle\tilde{\varphi}\rangle_y$. Incorrect implementation of `field-line-average-term`
Type	string
Default	`no-field-line-average-term`

3.2.8 The `time_advance_knobs` namelist

3.2.9 The `kt_grids_knobs` namelist

3.2.10 The `kt_grids_range_parameters` namelist

3.2.11 The `init_g_knobs` namelist

3.2.12 The `knobs` namelist

3.2.13 The `species_knobs` namelist

3.2.14 The `species_parameters_#` namelist

3.2.15 The `stella_diagnostics_knobs` namelist

3.2.16 The `reinit_knobs` namelist

3.2.17 The `layouts_knobs` namelist

3.2.18 The `neoclassical_input` namelist

3.2.19 The `sfincs_input` namelist

Chapter 4

Output

In this chapter we list the output files of the code and their content.

Chapter 5

Diagnostics

In this chapter the diagnostics tools available for reading, diagnosing and postprocessing the output delivered by `stella` are described.

5.1 The output files

5.2 Diagnosing at the `python` prompt

5.3 The `stella` GUI

Bibliography