eTF code

(extended Two-Fluid model)

Documentation

Silvio Sergio Cerri

Features of currently released version

Version 1.0:

➢ parallel (**MPI**) code
➢ two-dimensional (**2D**) equations
➢ **open** boundary conditions along x
➢ **periodic** boundary conditions along y

➢ generalized Ohm's law with: **Hall term** (JxB), **thermo-electric effect** (div[P_e])
➢ **anisotropic pressures** for both species (protons and electrons)
➢ 1st-order **finite-Larmor-radius (FLR) corrections** of the ions (assumes B mainly along z)
➢ **double-adiabatic closure** (zero heat fluxes)
➢ **massless electrons** (no electron-inertia effects)

➢ default initial condition: FLR-corrected **shear-flow layer** (e.g., Kelvin-Helmholtz instability)

eTF model equations

reference:

Cerri *et al.*, Physics of Plasmas **20**, 112112 (2013)

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{U}) = 0, \tag{1}$$

$$\frac{\partial (n\mathbf{U})}{\partial t} + \nabla[n\mathbf{U}\mathbf{U} + \mathbf{\Pi}_\mathbf{B} + \mathbf{\Pi}_e^{(0)} + \mathbf{\Pi}_i^{(0)} + \mathbf{\Pi}_i^{(1)}] = 0, \tag{2}$$

$$\mathbf{E} = -\mathbf{U} \times \mathbf{B} + \frac{\mathbf{J} \times \mathbf{B}}{n} - \frac{\nabla \mathbf{\Pi}_e^{(0)}}{n}, \tag{3}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}; \quad \nabla \times \mathbf{B} = \mathbf{J}, \tag{4}$$

$$\frac{\partial p_{i\perp}}{\partial t} + \nabla(p_{i\perp}\mathbf{U}) = -p_{i\perp}(\mathbf{I} - \mathbf{bb}) : \nabla\mathbf{U} - \mathbf{\Pi}_i^{(1)} : \nabla\mathbf{U}, \tag{5}$$

$$\frac{\partial p_{e\perp}}{\partial t} + \nabla\left[p_{e\perp}\left(\mathbf{U} - \frac{\mathbf{J}}{n}\right)\right] = -p_{e\perp}\left[(\mathbf{I} - \mathbf{bb}) : \nabla\left(\mathbf{U} - \frac{\mathbf{J}}{\mathbf{n}}\right)\right], \tag{6}$$

$$\frac{\partial p_{i\parallel}}{\partial t} + \nabla(p_{i\parallel}\mathbf{U}) = -2p_{i\parallel}\,\mathbf{bb} : \nabla\mathbf{U}, \tag{7}$$

$$\frac{\partial p_{e\parallel}}{\partial t} + \nabla\left[p_{e\parallel}\left(\mathbf{U} - \frac{\mathbf{J}}{n}\right)\right] = -2p_{e\parallel}\left[\mathbf{bb} : \nabla\left(\mathbf{U} - \frac{\mathbf{J}}{n}\right)\right], \tag{8}$$

where $\mathbf{\Pi}_\mathbf{B} = (B^2/2)(\mathbf{I} - \mathbf{bb}) - (B^2/2)\mathbf{bb}$, $\mathbf{\Pi}_\alpha^{(0)} = p_{\alpha\perp}(\mathbf{I} - \mathbf{bb}) + p_{\alpha\parallel}\mathbf{bb}$. Here $p_{\alpha\parallel}$ and $p_{\alpha\perp}$ are the parallel and perpendicular components of the magnetic pressure tensor of the α species, $\mathbf{b} = \mathbf{B}/B$ is the unit vector along the local magnetic field, and $\mathbf{I} - \mathbf{bb}$ is the projector in the perpendicular plane. Finally $\mathbf{\Pi}_i^{(1)}$ is the first order ion gyroviscosity tensor. In the strong magnetic guide field limit, in our case $\mathbf{B} \simeq B\mathbf{e}_z$, the ion gyroviscosity tensor components $\Pi_{i,lm}^{(1)}$, in dimensionless form, are given by[9]

$$\Pi_{i,zz}^{(1)} = 0, \tag{9}$$

$$\Pi_{i,xx}^{(1)} = -\Pi_{i,yy} = -\frac{1}{2}\frac{p_{i\perp}}{B}(\partial_y u_{i,x} + \partial_x u_{i,y}), \tag{10}$$

$$\Pi_{i,xy}^{(1)} = \Pi_{i,yx}^{(1)} = -\frac{1}{2}\frac{p_{i\perp}}{B}(\partial_y u_{i,y} - \partial_x u_{i,x}), \tag{11}$$

$$\Pi_{i,xz}^{(1)} = \Pi_{i,zx}^{(1)} = -\frac{1}{B}[(2p_{i\parallel} - p_{i\perp})\partial_z u_{i,y} + p_{i\perp}\partial_y u_{i,z}] - \frac{1}{B}\partial_y q_{i\perp}, \tag{12}$$

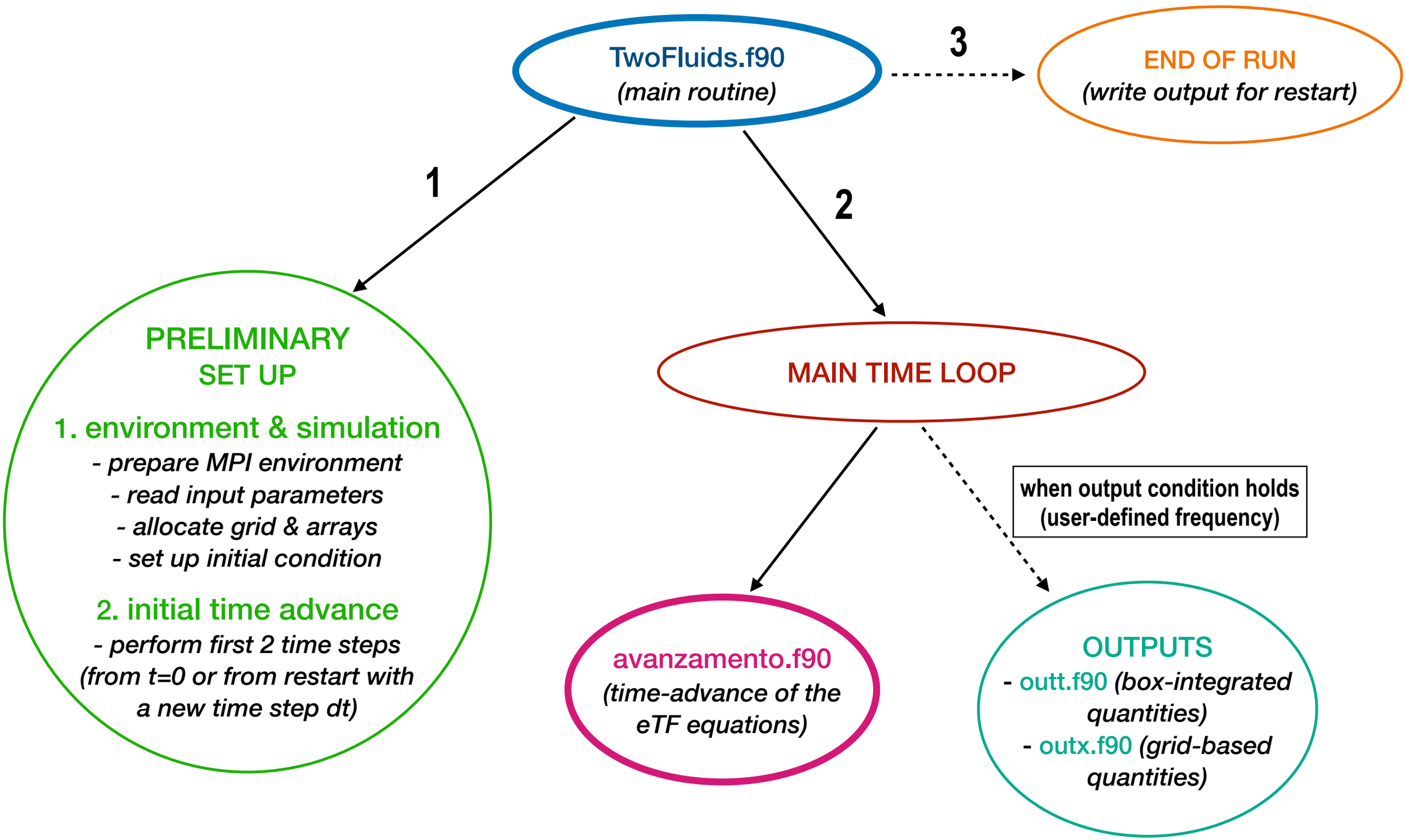
$$\Pi_{i,yz}^{(1)} = \Pi_{i,zy}^{(1)} = \frac{1}{B}[(2p_{i\parallel} - p_{i\perp})\partial_z u_{i,x} + p_{i\perp}\partial_x u_{i,z}] + \frac{1}{B}\partial_x q_{i\perp}, \tag{13}$$

where $q_{i\perp}$ and $q_{i\parallel}$ are the heat fluxes along the magnetic field of the perpendicular and parallel thermal ion energy (hereafter neglected, as in pressure equations).

➢ equations are normalized using *ion-inertial length*, *ion-cyclotron frequency*, and *Alfvén speed*:

$$d_i,\ \Omega_i,\ V_A$$

Schematic structure of the code



How to compile (1)

Load needed modules:

1. **intel**
2. **intel-mpi**
3. **intel-mkl**

example: on **Perseus** cluster @ Princeton
(https://researchcomputing.princeton.edu/systems-and-services/available-systems/perseus)

```
> module load intel
> module load intel-mpi
> module load intel-mkl
```

```
Currently Loaded Modulefiles:
  1) intel-mkl/2020.1/1/64       2) intel/19.1/64/19.1.1.217    3) intel-mpi/intel/2019.7/64
```

How to compile (2)

Check available *compilers* and add desired *optimization flags* in “**Machine**” file

Default configuration:

```
LIB=
FFLAGS= -O3 -c -I modules
F90FLAGS= -O3 -c -I modules
LNKFLAGS=
F77= mpif90
F90= mpif90
LNK= mpif90
```

How to compile (3)

Actual compilation:

First, use the command “**make veryclean**” to clean up before any new compilation.
(recommended over “make clean”)

Then, use “**make**” to compile the code.

Successful compilation will produce a “**.x**” executable

Default exectuable name: **eTF.x**
(see “**makefile**” for further details and/or to change executable name)

Simulation parameters (1)

BEFORE compilation:

☞ grid size: (Nx, Ny, Nz)

this is defined in **`TF_mod.f90`** (in `paramerer_mod`)

```
INTEGER, PARAMETER :: nx=4096, ny=2048, nz=1
```

Simulation parameters (2)

AFTER compilation:

Input parameters are given through the file **`2fl.com`**

(see **`condinit.f90`** for definition and variables involved in the intial plasma configuration)

Questions and bug reports

Contact:

Silvio Sergio Cerri (scerri@astro.princeton.edu)

Department of Astrophysical Sciences, Princeton University