BOUT, HESEL, FELTOR comparison

May 28, 2020

Abstract

This document is the technical documentation of three different implementations of the same model.

1 Model

1.1 Equations

n is the electron density and ω the vorticity density. ϕ is the electric potential. We use Cartesian coordinates x, y. Gyro-Bohm normalization. κ is the curvature, ν diffusivity.

$$\nabla^2\phi = \omega, \tag{1a}$$

$$\frac{\partial n}{\partial t} = \{n,\phi\} + \kappa\frac{\partial \phi}{\partial y} + \nu\nabla^2 n \tag{1b}$$

$$\frac{\partial \omega}{\partial t} = \{\omega,\phi\} - \kappa\frac{\partial n}{\partial y} + \nu\nabla^2 \omega \tag{1c}$$

For the Hasegawa - Wakatani model we the following equations

$$\nabla^2\phi = \omega, \tag{2a}$$

$$\frac{\partial n}{\partial t} = -g\frac{\partial \phi}{\partial y} + \alpha(\tilde{\phi} - \tilde{n}) + \{n,\phi\} + D\nabla^2 n \tag{2b}$$

$$\frac{\partial \omega}{\partial t} = \alpha(\tilde{\phi} - \tilde{n}) + \{\omega,\phi\} + \mu\nabla^2 \omega \tag{2c}$$

The variables $\tilde{\phi}$ and $\tilde{n}$ represents low frequency fluctuations. Their relation with ϕ and n is given by

$$\phi = \tilde{\phi} + < \phi > \tag{3a}$$

$$n = \tilde{n} + < n > \tag{3b}$$

where $< ... >$ represents the averagive in the poloidal direction. For this reason, we can write the HW model equations as

$$\nabla^2\phi = \omega, \tag{4a}$$

$$\frac{\partial n}{\partial t} = -g\frac{\partial \phi}{\partial y} + \alpha[(\phi - <\phi>) - (n - <n>)] + \{n, \phi\} + D\nabla^2 n \tag{4b}$$

$$\frac{\partial \omega}{\partial t} = \alpha[(\phi - <\phi>) - (n - <n>)] + \{\omega, \phi\} + \mu\nabla^2\omega \tag{4c}$$

which can be approximated with $<n> \approx 0$ and $<\phi> \approx 0$.

1.2 Domain, boundary and initial conditions

The domain is $[0, l_x] \times [0, l_y]$. We have periodic boundaries in y. In x we choose homogeneous Dirichlet boundary conditions for all quantities. Initialization of n is a Gaussian

$$n(x, y) = A\exp\left(-\frac{(x - X)^2 + (y - Y)^2}{2\sigma^2}\right) \tag{5}$$

where $X = p_x l_x$ and $Y = p_y l_y$ with $p_x,\, p_y \in [0, 1]$ are the initial centre of mass position coordinates, A is the amplitude and σ the radius of the blob. We initialize

$$\omega = \phi = 0 \tag{6}$$

1.3 Invariants

Mass, free energy and total energy are

$$M(t) := \int n$$

$$F(t) := \int n^2/2 + (\nabla\phi)^2/2$$

$$H(t) := \int -\kappa n x + (\nabla\phi)^2/2$$

We have

$$\frac{\partial}{\partial t}M = \nu\int \nabla^2 n \tag{7}$$

$$\frac{\partial}{\partial t}F = \nu\int n\nabla^2 n - \phi\nabla^2\omega \tag{8}$$

$$\frac{\partial}{\partial t}H = -\nu\int \kappa x\nabla^2 n + \phi\nabla^2\omega \tag{9}$$

2 Numerical methods

2.1 Feltor

discontinuous Galerkin on structured grid

Term	Method	Description
coordinate system	Cartesian 2D	equidistant discretization of $[0, l_x] \times [0, l_y]$, equal number of Gaussian nodes in x and y
matrix inversions	multigrid with conjugate gradient	Use previous two solutions to extrapolate initial guess
$\boldsymbol{E} \times \boldsymbol{B}$ advection	Arakawa	s.a. [1]
curvature terms	direct	flux conserving
time	Karniadakis multistep	$3rd$ order explicit, diffusion $2nd$ order implicit

3 Compilation and useage

3.1 Feltor

There are three programs convection.cpp, convection_hpc.cpp and convection_mpi.cpp . Compilation with

```
make device = <omp or gpu>
```

Run with

```
convection input.json
convection_hpc input.json output.nc
echo np_x np_y | mpirun -n np_x*np_y convection_mpi\
    input.json output.nc
```

All programs write performance informations to std::cout. The first opens a terminal window with live simulation results. The other two write the results to disc. The second is for shared memory systems. The third for distributed memory systems, which expects the distribution of processes in the x and y directions.

3.1.1 Input file structure

Input file format: json

Name	Type	Example	Default	Description
n	integer	3	-	# Gaussian nodes in x and y

Nx	integer	60	-	# grid points in x
Ny	integer	60	-	# grid points in y
dt	integer	4.0	-	time step in units of c_s/ω_s
n_out	integer	3	-	# Gaussian nodes in x and y in output
Nx_out	integer	60	-	# grid points in x in output fields
Ny_out	integer	60	-	# grid points in y in output fields
itstp	integer	10	-	steps between outputs
maxout	integer	100	-	# outputs excluding first
eps_pol	float	1e-6	-	accuracy of polarisation solver
eps_time	float	1e-10	-	accuracy of implicit time-stepper
stages	integer	3	3	# of stages in the multigrid algorithm (Nx and Ny have to be divisable by 2**stages)
curvature	float	0.00015	-	magnetic curvature κ
nu_perp	float	5e-3	-	pependicular viscosity ν
amplitude	float	0.5	-	amplitude A of the blob
sigma	float	10	-	blob radius σ
posX	float	0.3	-	blob x-position in units of l_x, i.e. $X = p_x l_x$
posY	float	0.5	-	blob y-position in units of l_y, i.e. $Y = p_y l_y$
lx	float	200	-	l_x
ly	float	200	-	l_y
bc_x	char	”DIR”	-	boundary condition in x (one of PER, DIR, NEU, DIR_NEU or NEU_DIR)
bc_y	char	”PER”	-	boundary condition in y (one of PER, DIR, NEU, DIR_NEU or NEU_DIR)
rows	integer	2	2	# of rows in window for live-plot
cols	integer	1	1	# of cols in window for live-plot
width	integer	500	500	width of the window in pixel in live-plot
height	integer	1000	1000	height of the window in pixel in live-plot

The default value is taken if the value name is not found in the input file. If there is no default and the value is not found, the program exits with an error message.

3.1.2 Structure of output file

Output file format: netcdf-4/hdf5

Name	Type	Dimension	Description

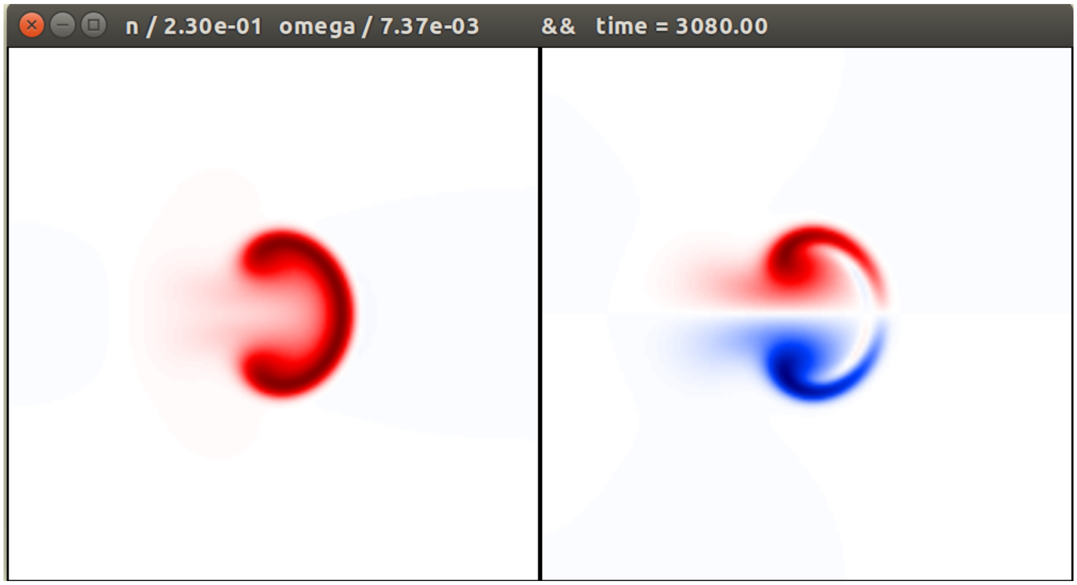
inputfile	text attribute	1	verbose input file as a string
energy_time	Dataset	1	timesteps at which 1d variables are written
time	Dataset	1	time at which fields are written
x	Dataset	1	x-coordinate
y	Dataset	1	y-coordinate
electrons	Dataset	3 (time, y, x)	electon density n
ions	Dataset	3 (time, y, x)	ion density N or vorticity density ω
potential	Dataset	3 (time, y, x)	electric potential ϕ
vorticity	Dataset	3 (time, y, x)	Laplacian of potential $\nabla^2\phi$
mass	Dataset	1 (energy_time)	$\int \mathrm{dV}\, n$
entropy	Dataset	1 (energy_time)	$\int \mathrm{dV}\, n^2/2$
kinetic	Dataset	1 (energy_time)	$\int \mathrm{dV}\, (\nabla\phi)^2//2$
curvature	Dataset	1 (energy_time)	$-\int \mathrm{dV}\, \kappa n x$
mass_diss	Dataset	1 (energy_time)	$\nu \int \mathrm{dV}\, \nabla^2 n$
entropy_diss	Dataset	1 (energy_time)	$\nu \int \mathrm{dV}\, n\nabla^2 n$
kinetic_diss	Dataset	1 (energy_time)	$-\nu \int \mathrm{dV}\, \phi\nabla^2\omega$
curvature_diss	Dataset	1 (energy_time)	$-\nu \int \mathrm{dV}\, \kappa x\nabla^2 n$

4 Diagnostics

4.1 Feltor

References

[1] L. Einkemmer, M. Wiesenberger, Comput. Phys. Comm. **185** (2014) 2865-2873



n / 2.30e-01 omega / 7.37e-03 && time = 3080.00

Figure 1: Example blob with parameters as in the example input file.