Mesh generation and simulation geometries

Ben Dudson and the BOUT++ team

York Plasma Institute, Department of Physics,
University of York, Heslington, York YO10 5DD, UK

BOUT++ Workshop, LLNL

17th December 2015
Filaments / blobs
- Transport of heat and particles in SOL
- 2D examples run on a laptop

Edge turbulence
- Formation of blobs
- Near SOL heat transport ($\lambda_q$)

Divertor simulations
- Spreading of particle and power fluxes to surfaces
- Interaction with neutral gas and detachment

Pedestal physics and ELMs
- Gyro-fluid models to capture FLR, Landau damping, and drift resonance effects
- L-H transition physics
- Stability and nonlinear dynamics
See tools/ sub-directory

- Analytic expressions
- Simplified grid generators e.g. flux tube (Cyclone), theta pinch, shifted circle tokamak
- TEQ (DSKGATO), ELITE formats
- EFIT equilibria (g-file)
- VMEC ... coming soon!
Up to 3 dimensions \((x, y, z)\)

Grid spacing \(dx(x, y), dy(x, y), dz\)
Coordinates

- Up to 3 dimensions \((x, y, z)\)
- Grid spacing \(dx(x, y), dy(x, y), dz\)
- By convention \(y\) is the parallel direction
  - Some operators don’t care:
    \[ \mathbf{v} = \nabla f \quad f = \nabla \cdot \mathbf{v} \quad \mathbf{u} = \nabla \times \mathbf{v} \]
  - Some operators assume \(\mathbf{B} = \nabla z \times \nabla x = \frac{1}{j} \mathbf{e}_y\)
    \[ f = [\phi, g] \quad f = \nabla_{\parallel} g = \nabla \cdot (\mathbf{b} g) \quad f = \partial_{\parallel} g = \mathbf{b} \cdot \nabla g \]
- Up to 3 dimensions \((x, y, z)\)
- Grid spacing \(dx(x, y), dy(x, y), dz\)
- By convention \(y\) is the parallel direction
- Mapping between curved grid and logical rectangle determines metric tensor
Coordinates

- Up to 3 dimensions \((x, y, z)\)
- Grid spacing \(dx(x, y), dy(x, y), dz\)
- By convention \(y\) is the parallel direction
- Mapping between curved grid and logical rectangle determines metric tensor

Metric tensors are functions of \(x\) and \(y\)
→ In the “standard” tokamak geometry \(z\) is a symmetry angle (toroidal angle)
- The grid is always periodic in \(z\) (for now)
The easiest way to specify a mesh is in the input file (BOUT.inp)

```
[mesh]
nx = 260  # Note 4 guard cells in X
ny = 1
nz = 256
```

This will produce a mesh with:

- Uniform grid spacing $dx = dy = dz = 1$
- Cartesian identity matrix as metric tensor
The easiest way to specify a mesh is in the input file (BOUT.inp)

```
[mesh]
nx = 260  # Note 4 guard cells in X
ny = 1
nz = 256

dx = 0.2
dz = 0.2
```

This will produce a mesh with:

- Uniform grid spacing as specified
- Cartesian identity matrix as metric tensor
The easiest way to specify a mesh is in the input file (BOUT.inp)

```
[mesh]
nx = 260  # Note 4 guard cells in X
ny = 1
nz = 256

Lx = 51.2
Lz = Lx

dx = Lx / (nx - 4)
dz = Lz / nz
```

This will produce a mesh with:

- Uniform grid spacing as specified
- Cartesian identity matrix as metric tensor
Analytic expressions can use:

- Normalised index space coordinates $0 \leq x \leq 1$, $0 \leq y \leq 2\pi$, and $0 \leq z \leq 2\pi$,
- Constants $\pi$ and $e$
- Common mathematical functions like $\sin$, $\cos$, $\tanh$, $\text{erf}$

```
[mesh]
...
Rmin = 5e-3  # minimum radius in meters
Rmax = 2.5e-2  # maximum radius
Rxy = Rmin + (Rmax - Rmin) * x
dr = (Rmax - Rmin) / (nx - 4)

dx = Bp * Rxy * dr
dy = length / ny
```
Example: 1-D modelling of divertor detachment

B.Dudson, B.Lipschultz

The SD1D model is being developed to study detachment dynamics and stability

- Fluid equations for plasma and neutrals
- Requires highly nonuniform grid

```
[mesh]

ny = 400
length = 25

dymin = 0.02
dy = (length / ny) * (1 + (1-dymin)*(1-y/pi))
```
Analytic meshes make convergence (MMS) testing easier
More complicated geometries don’t have “nice” analytic form
Require grid input file for e.g. experimental profiles:

```python
grid = "inputgrid.nc"
```

See conduction example:
- `examples/conduction/data/BOUT.inp`: Analytic input
- `examples/conduction/fromfile/BOUT.inp`: From grid file
  `(conduct_grid.nc)`
Grid files can be generated from scratch:

- Sheared slab: tools/slab/slab.pro
- Cyclone flux tube: tools/tokamak_grids/cyclone
- Shifted circle equilibria: tools/tokamak_grids/shifted_circle

Or from existing equilibria:

- DSKGATO (‘t’) files: tools/tokamak_grids/gato/
- ELITE input format: tools/tokamak_grids/elite/
- See README in tools/tokamak_grids/shifted_circle
- Two steps: Generate mesh; process to create BOUT++ input
The grid used for field-aligned X-point simulations consists of several blocks

- Every processor in BOUT++ has a logically rectangular grid
Branch cuts and communications

The grid used for field-aligned X-point simulations consists of several blocks:

- Every processor in BOUT++ has a logically rectangular grid.
- Connections between blocks are done by changing communications between processors.

![Diagram showing grid layout and connections between blocks.](image-url)
The grid used for field-aligned X-point simulations consists of several blocks:

- Every processor in BOUT++ has a logically rectangular grid.
- Connections between blocks are done by changing communications between processors.
The locations of these branch cuts are controlled by quantities in the input mesh:

- **ixseps1, ixseps2**: Separatrix radial locations (number of points inside each separatrix)
- **jyseps1_1, jyseps2_1, jyseps1_2, jyseps2_2**: Poloidal (y) branch cut locations
Create a field-aligned mesh from EFIT G-EQDSK files
Largely automatic, given required $\psi$ ranges and resolution
Uses mainly heuristic methods
Written in IDL
A partial translation to Python also exists
  tools/tokamak_grids/pyGridgen (thanks to G.Breyiannis)
Hypnotoad grid generator
tools/tokamak_grids/gridgen

Mesh | Profiles | Output
--- | --- | ---
Read G-EQDISK
Restore R-Z
Read boundary
Radial points: 36
Poloidal points: 54
Inner psi: 0.900000
Outer psi: 1.100000
Sep. spacing: 1
Par. vs pol: 0.000000
Ypt dist x: 1
Generate mesh
Detailed settings
Save state
Restore state
Strict boundaries
Simplify boundary
Single radial grid
Fast
Nonorthogonal mesh

Successfully read /hwdisks/home/bd512/bout-master/tools/tokamak_grids
Hypnotoad grid generator
tools/tokamak_grids/gridgen

Ben Dudson, YPI

BOUT++ Workshop 2015 (16 of 25)
Hypnotoad grid generator
tools/tokamak_grids/gridgen

Successfully generated mesh. All glory to the Hypnotoad!
Hypnotoad grid generator
tools/tokamak_grids/gridgen

Successfully generated mesh. All glory to the Hypnotoad!

J_par from f1, Solid from f1 and J_par at y=46 Solid from f1, respectively.
Hypnotoad grid generator
tools/tokamak_grids/gridgen

Successfully generated mesh. All glory to the Hypnotoad!
The mesh can be processed to keep only some domains.

N.Walkden studying divertor transport.

N.Walkden: Collisional electron transport in MAST.

Submitted to PPCF (2015)
The mesh can be processed to keep only some domains.

N.Walkden studying divertor transport.

Strong enhancement of collisional (Braginskii) transport needed to match experiment.

N.Walkden: Collisional electron transport in MAST.

Submitted to PPCF (2015)
The mesh can be processed to keep only some domains

N.Walkden studying divertor transport

Strong enhancement of collisional (Braginskii) transport needed to match experiment

Significant Bohm-like diffusion can reproduce reasonable profiles

N.Walkden: Collisional electron transport in MAST.

Submitted to PPCF (2015)
Coming soon...
Non-orthogonal meshes
J. Leddy, B. Shanahan, N. Walkden, B. Dudson

- The standard mesh and coordinate system is orthogonal in the poloidal plane
- Does not conform to divertor surfaces
- Produces large variation in cell spacing around X-point
Non-orthogonal meshes
J.Leddy, B.Shanahan, N.Walkden, B.Dudson
\( \eta = \text{const} \quad \eta \neq \text{const} \)
Core-Edge coupling with BOUT++ and CENTORI

Sharing via an overlap region ("handshake" method)
Core-Edge coupling with BOUT++ and CENTORI
Sharing via an overlap region ("handshake" method)
Profiles evolve self-consistently
Fluctuations from the core propagate to target
Non-orthogonal meshes: Divertor leg neutrals
J.Leddy, B.Dudson

- Interaction of plasma with fluid neutrals (Navier Stokes), and coupling to EIRENE
- Impact of divertor angle on divertor heat loads and detachment
An alternative approach to parallel derivatives

Grid points are not aligned on magnetic field

Coordinate system can be Cartesian, cylindrical. **No singularity at X-point**

Follow field-lines to neighboring planes and interpolate

Magnetic field needs to be locally integrable, but not globally. **No assumption of flux surfaces**
Implemented in BOUT++
Convergence tested using MMS
- Implemented in BOUT++
- Convergence tested using MMS
- Straight stellarator geometry
- Cartesian mesh in $x - z$, FCI in $y$
Flux-Coordinate Independent (FCI)

- Implemented in BOUT++
- Convergence tested using MMS
- Straight stellarator geometry
- Cartesian mesh in $x - z$, FCI in $y$
- Diffusion equation to test numerical diffusion

$$\partial_t f = \nabla^2_{\parallel} f$$
Conclusions

- BOUT++ geometry and topology are quite flexible
- Retain some limitations from tokamak simulation origins
- Many scripts and examples to get started in a range of geometries

- Ongoing efforts to improve schemes for tokamak and more complicated geometries