Curvilinear and flux coordinates in BOUT++

A.Smolyakov, O.Chapurin

November 1, 2016

Contents

1 Introduction

BOUT++ [1] is designed to simulate arbitary number of plasma fluid equations in 3D curvilinear coordinates using finite-difference methods. It has been developed from BOUT [3] simulation code for 2-fluid tokamak edge simulations, from where it inherited coordinate system metric tensor $g^{ij} = g^{ij}(x, y)$ (constant in one dimension). It means, it is restricted to the coordinate system with axi- or translationally symmetric geometry. Even 2D metric tensors allows the code to be used to simulate plasmas cylindrical or non-orthogonal coordinate systems such as flux coordinates for tokamak simulations.

2 Coordinate system and operators

To specify a coordinate system one should define each component of the metric tensors g^{ij} and g_{ij}. Jacobian J and Crystoffel symbol Γ^i_{jk} components will be calculated from given metric tensor components, but can be specified as well. All differential operators contain these quantities. By default metric is cartesian $g^{ij} = g_{ij} = I$, where I is the identity matrix.

We can specify these quantities in either grid file or input option (BOUT.inp) file, from where they are read on a program initialization step. Another option is to define them in the physics module, which could lead to the implementation of evolving in time (space) metric tensors. As it was mentioned, our coordinate system is restricted to having one symmetry direction (z), so all metric tensor components are 2D fields. Note that grid spacing $dx(x, y), dy(x, y)$ are 2D fields as well, but dz is a scalar, so we can work with non-uniform meshes in x-, y-directions only.

By convention y coordinate is the parallel direction. Thus, we can divide differential operators into two categories: those who are independent of any coordinate system and those which assume $\mathbf{B} = \nabla z \times \nabla x$ in a Clebsch coordinate system, where $\mathbf{B}$ aligned with the y coordinate. The following table contains differential operators for general coordinate system:

Math expression	Result=Operator(Input)
$\mathbf{v} = \nabla f$	Vector = Grad(Field)
$f = \nabla \cdot \mathbf{a}$	Field = Div(Vector)
$\mathbf{v} = \nabla \times \mathbf{a}$	Vector = Curl(Vector)
$f = \mathbf{a} \cdot \nabla g$	Field = `V_dot_Grad`(Vector, Field)
$\mathbf{v} = \mathbf{a} \cdot \nabla \mathbf{b}$	Vector = `V_dot_Grad`(Vector, Vector)
$f = \nabla^2 g$	Field = Laplacian(Field)

General expression for these operators are:

$$\nabla f = \frac{\partial f}{\partial u^i} \nabla u^i, \tag{1}$$

$$\nabla \cdot \mathbf{A} = \frac{1}{J} \frac{\partial}{\partial u^i} \left(J \sqrt{g^{ij}} A_j \right), \tag{2}$$

$$\nabla^2 f = G^j \frac{\partial f}{\partial u^i} + g^{ij} \frac{\partial^2 f}{\partial u^i \partial u^j}, \tag{3}$$

where

$$G^j = \frac{1}{J} \frac{\partial}{\partial u^i} \left(J g^{ij} \right). \tag{4}$$

Operators which assume that the equilibrium magnetic field is written in Clebsch form are:

Math expression	Result=Operator(Input)
$\partial^0_{\parallel} = \mathbf{b}_0 \cdot \nabla$	Scalar = `Grad_par`(Scalar)
$\nabla^0_{\parallel} f = B_0 \partial^0_{\parallel} \left(\frac{f}{B_0} \right)$	Scalar = `Div_par`(Scalar)
$\nabla^2_{\parallel} f = \nabla \cdot \mathbf{b}_0 \mathbf{b}_0 \cdot \nabla f$	Scalar = `Laplace_par`(Scalar)
$\nabla^2_{\perp} f = \nabla^2 f - \nabla^2_{\parallel} f$	Field = `Laplace_perp`(Field)
$f = \mathbf{b}_0 \cdot \nabla \phi \times \nabla A$	Scalar = `b0xGrad_dot_Grad`(Scalar, Scalar)

Here $\mathbf{B} = B_0 \mathbf{b}$ is a background equilibrium (unperturbed) magnetic field and

$$\mathbf{B} = \nabla z \times \nabla x = \frac{1}{J} \mathbf{e}_y, \tag{5}$$

according to Clebsch form. The unit vector $\mathbf{b}$ in the direction of equilibrium $\mathbf{B}$

$$\mathbf{b} = \frac{1}{JB_0} \mathbf{e}_y = \frac{1}{JB_0} \left(g_{xy} \nabla x + g_{yy} \nabla y + g_{zy} \nabla z \right) \tag{6}$$

3 Cylindrical coordinates

The easiest way to specify cylindrical coordinates is to use a known expression for a contravariant metric tensor components

$$g^{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/r^2 \end{pmatrix} \tag{7}$$

Due to orthogonality of cylindrical coordinates covariant form of metric tensor will be just $g_{ii} = 1/g^{ii},\ i = 1, 2, 3$. Here BOUT's (x, y, z) represent a cylindrical (r, z, ϕ).

This metric tensor should also work with (z, r, ϕ), i.e. BOUT's y is the radial direction, but I couldn't succeed with that and don't understand why.

3.1 Differential operators

Inverse Jacobian J^{-1} for given coordinates can be found using metric tensor $J^{-1} = \det g^{ij} = 1/r$, so $J = r$. Using Equations (1, 2), a general form of these operatoes for our cylidndrical coordinates

$$\nabla f = \frac{\partial f}{\partial r}\mathbf{e}_r + \frac{\partial f}{\partial r}\mathbf{e}_z + \frac{1}{r}\frac{\partial f}{\partial \phi}\mathbf{e}_\phi, \tag{8}$$

$$\nabla \cdot \mathbf{A} = \frac{1}{r}\frac{\partial}{\partial r}(rA_r) + \frac{\partial A_z}{\partial z} + \frac{1}{r}\frac{\partial A_\phi}{\partial \phi}. \tag{9}$$

From Eq. (4) we find that

$$G^1 = \frac{1}{r}\frac{\partial}{\partial r}(r) = \frac{1}{r},$$
$$G^2 = \frac{1}{r}\frac{\partial}{\partial z}(r) = 0,$$
$$G^3 = \frac{1}{r}\frac{\partial}{\partial \phi}\left(r\frac{1}{r}\right) = 0.$$

Finally, we have the general expression for Laplacian operator from Eq.(3)

$$\nabla^2 f = \frac{1}{r}\frac{\partial f}{\partial r} + \frac{\partial^2 f}{\partial r^2} + \frac{\partial^2 f}{\partial z^2} + \frac{1}{r^2}\frac{\partial^2 f}{\partial \phi^2}. \tag{10}$$

3.2 Verification with BOUT++

To verify this method a simple heat conduction problem was implemented[1] and solved

$$\frac{\partial T}{\partial t} = \chi \nabla^2 T, \tag{11}$$

with a following input options for geometry, according to metric tensor (7)

[1]More details (animation, source files) can be found in https://github.com/alxmar/conduction-cylinder

```
[mesh]  # Mapping to cylinder: x -> r,  y -> z,  z -> phi

nx = 68 # including 2*2 guard points
ny = 1

Rmin = 0.05
Rmax = 1.8

Rxy = Rmin + (Rmax - Rmin)*x    # Rxy = [Rmin, ..., Rmax]
dr = (Rmax - Rmin) / (nx-4)     # 2 guard points on each side

Lz = 1.0
dx = dr
dy = 1.0

### Contravariant metric tensor components
g11 = 1.0
g22 = 1.0
g33 = 1.0/Rxy^2

### Covariant metric tensor components
g_11 = 1 / g11
g_22 = 1 / g22
g_33 = 1 / g33
```

The analytical solution to this two-dimensional cylindrical (r, ϕ) problem with Neumann boundary conditions is:

$$T(r,\phi,t) = \sum_{m=0}^{\infty}\sum_{k=1}^{\infty} A_{m,k} J_m\left(\frac{B_{m,k} r}{r_0}\right)\cos(m\phi)\exp\left(-\chi\frac{B_{m,k}^2}{r_0^2}t\right) + C, \quad (12)$$

where $B_{m,k}$ is the k^{th} zero of the function $J_m^{'}(r)$, C is arbitary constant. With the intial condition $T_0(r, \phi, t = 0) = \cos(2\phi)$ (Fig.1) the solution is simplified to[2]

$$T(r,\phi,t) = \sum_{k=1}^{20} A_{2,k} J_2\left(\frac{B_{2,k} r}{r_0}\right)\cos(2\phi)\exp\left(-\chi\frac{B_{2,k}^2}{r_0^2}t\right), \quad (13)$$

where we can take first 20 terms of the infinite sum. Solving Eq.(11) in BOUT++ with a clyndrical metric components, presented above, we can see that the numerical and analytical results are highly consistent (Fig.2).

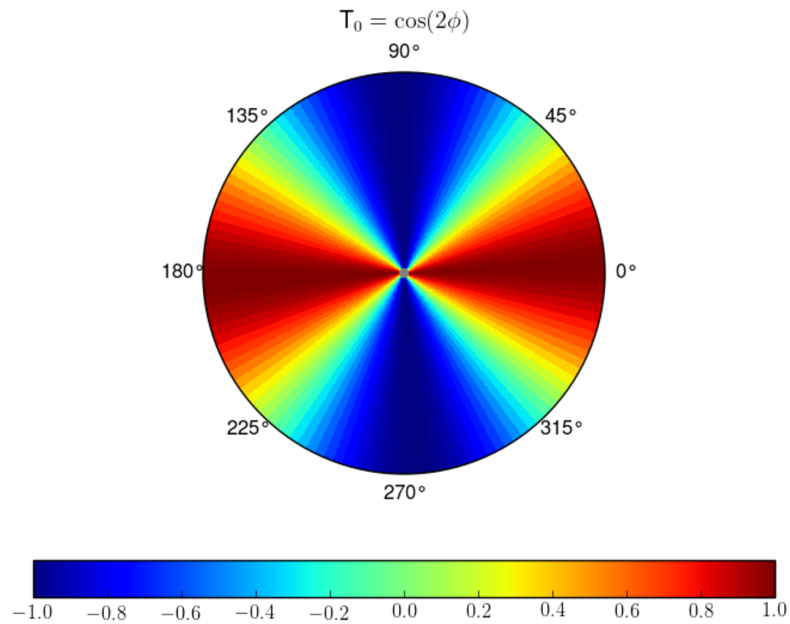
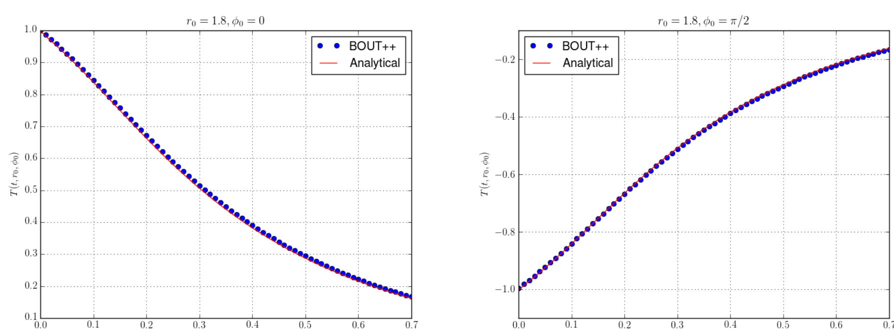


Figure 1: Plot in the polar coordinates of the initial condition for the 2-D heat conduction equation, $T_0(r, \phi, t = 0) = \cos(2\phi), n_r = 64, n_\phi = 128$.



(a) (b)

Figure 2: The comparsion between simulation results from BOUT++ and the analytical solution at two different positions

4 Cylindrical flux coordinates

4.1 Axial constant magnetic field (LAPD without mirrors)

Here we introduce an orthogonal cylindrical coordinate system (ψ, z, ϕ), where ψ is the axial flux, z is the axial coordinate, ϕ the azimuthal angle (from 0 to 2π). Clebsch form (5) for megnetic field implies that

$$\mathbf{B} = \nabla\phi \times \nabla\psi$$

Chosing a magnetic flux to be $\psi = B_0 r^2/2$, where r is the radial coordinate in the standard cylindrical coordinates, we obtain the axial constant magnetic field $\mathbf{B} = B_0 \mathbf{e}_z$. The contravariant metric tensor is defined as

$$g^{ij} \equiv \mathbf{e}^i \cdot \mathbf{e}^j = \nabla u^i \cdot \nabla u^j, \tag{14}$$

where unit vectors for our coordinates are

$$\begin{aligned} \mathbf{e}^{\psi} &\equiv \nabla\psi = B_0 r \hat{\mathbf{e}}_{\mathbf{r}}, & (15)\\ \mathbf{e}^{z} &\equiv \nabla z = \hat{\mathbf{e}}_{\mathbf{z}}, & (16)\\ \mathbf{e}^{\phi} &\equiv \nabla\phi = \frac{1}{r}\hat{\mathbf{e}}_{\phi}, & (17) \end{aligned}$$

Therefore we have

$$g^{ij} = \begin{pmatrix} (B_0 r)^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/r^2 \end{pmatrix}$$

and a covariant form

$$g_{ij} = \begin{pmatrix} 1/(B_0 r)^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r^2 \end{pmatrix}$$

4.1.1 Differential operators

(this subsubsection is not finished)

The Jacobian is found to be constant $J = B_0$.

Laplacian operator in general is given by

$$\begin{aligned}
\nabla^2 F &= \frac{1}{J}\frac{\partial}{\partial u^i}\left(Jg^{ij}\frac{\partial F}{\partial u^j}\right) = \frac{\partial}{\partial x^B}\left(g^{xx}\frac{\partial F}{\partial x^B}\right) + \frac{\partial}{\partial y^B}\left(g^{yy}\frac{\partial F}{\partial y^B}\right) + \frac{\partial}{\partial z^B}\left(g^{zz}\frac{\partial F}{\partial z^B}\right) \\
&= \frac{\partial}{\partial x^B}\left(B_0^2 r^2\frac{\partial F}{\partial x^B}\right) + \frac{\partial}{\partial y^B}\left(\frac{\partial F}{\partial y^B}\right) + \frac{\partial}{\partial z^B}\left(\frac{1}{r^2}\frac{\partial F}{\partial z^B}\right) \\
&= \frac{\partial}{\partial \psi}\left(B_0^2 r^2\frac{\partial F}{\partial \psi}\right) + \frac{\partial}{\partial z}\left(\frac{\partial F}{\partial z}\right) + \frac{\partial}{\partial \phi}\left(\frac{1}{r^2}\frac{\partial F}{\partial \phi}\right) \\
&= \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial F}{\partial r}\right) + \frac{\partial}{\partial z}\left(\frac{\partial F}{\partial z}\right) + \frac{\partial}{\partial \phi}\left(\frac{1}{r^2}\frac{\partial F}{\partial \phi}\right),
\end{aligned}$$

where we have used a transformation

$$\frac{\partial}{\partial \psi} = \frac{1}{B_0 r}\frac{\partial}{\partial r}.$$

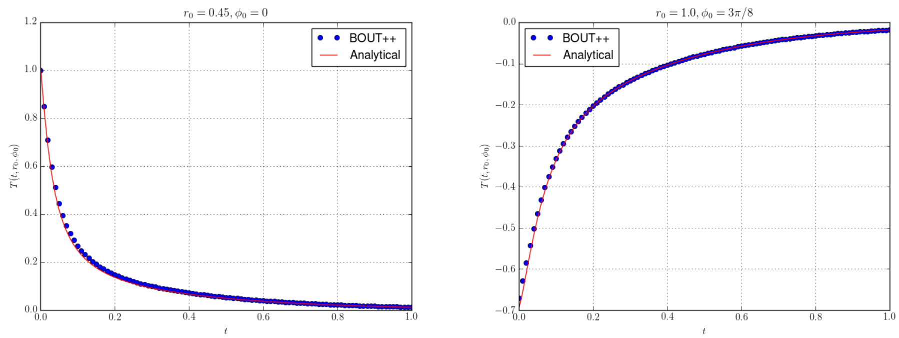
Therefore, it becomes a standard Laplacian operator in cylindrical coordinates.

4.1.2 Verification with BOUT++

$$\frac{\partial T}{\partial t} = \chi \nabla^2 T \tag{18}$$

with the input file:

```
[mesh]  # Mapping: x -> r,   y -> z,   z -> phi



nx = 36
ny = 1

#hthe = 3.0
Rmin = 0.05
Rmax = 1.8

Rxy = Rmin + (Rmax - Rmin)*x    # Rxy = [Rmin, ..., Rmax]
dr = (Rmax - Rmin) / (nx-4)

Bxy = 1.
Ly = 3.0
dy = 1.
dx = Bxy*Rxy*dr # Required by definition of flux psi = Bxy*r^2/2

### Contravariant metric tensor components
g11 = (Bxy*Rxy)^2
```



(a) (b)

Figure 3: The comparsion between simulation results from BOUT++ and the analytical solution at two different positions

```
g22 = 1.0
g33 = 1./(Rxy)^2

### Covariant metric tensor components
g_11 = 1 / g11
g_22 = 1 / g22
g_33 = 1 / g33

J = Bxy
```

Intial conditions and other parameters were identical to the heat conduction problem in the previous section. Results are presented in Fig.(3).

4.2 Radial constant magnetic field

Here we introduce an orthogonal cylindrical coordinate system (ψ, r, ϕ), where ψ is the radial flux, r is the radial coordinate, ϕ the azimuthal angle (from 0 to 2π). Clebsch form (5) for megnetic field implies that

$$\mathbf{B} = \nabla\phi \times \nabla\psi$$

Chosing a magnetic flux to be $\psi = B_0 z$, where z is the axial coordinate in the standard cylindrical coordinates, we obtain the radial constant magnetic field

$$\mathbf{B} = \frac{B_0}{r}\mathbf{e}_r. \tag{19}$$

Unit vectors for our coordinates are

$$
\begin{aligned}
\mathbf{e}^{\psi} &\equiv \nabla\psi = B_0\mathbf{e}_z, & (20)\\
\mathbf{e}^{z} &\equiv \nabla r = \mathbf{e}_r, & (21)\\
\mathbf{e}^{\phi} &\equiv \nabla\phi = \frac{1}{r}\mathbf{e}_\phi, & (22)
\end{aligned}
$$

Therefore we have

$$
g^{ij} = \begin{pmatrix} B_0^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/r^2 \end{pmatrix}
$$

and a covariant form

$$
g_{ij} = \begin{pmatrix} 1/B_0^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r^2 \end{pmatrix}
$$

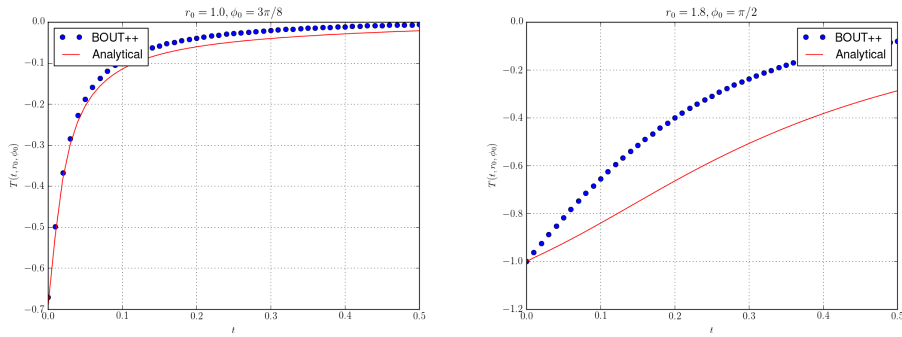
With $B_0 = 1$ this metric tensor appeared to be the same as in (7), but here we have BOUT's x representing z, and we know that it is working, but for (r, z, ϕ), as it was shown. Even dimensions of ψ $B_0 z$ are not magnetic flux dimensions, so proabably this is wrong.

4.2.1 Verification with BOUT++

$$
\frac{\partial T}{\partial t} = \chi\nabla^2 T \tag{23}
$$

with the input file:

```
[mesh]  # Mapping to cylinder: x -> z,  y -> r,  z -> phi

nx = 5 # including 2*2 ghost points
ny = 128

Rmin = 0.05
Rmax = 1.8

Rxy = Rmin + (Rmax - Rmin)*y/(2*pi)   # as y is normalized to 0,
    2*pi
dr = (Rmax - Rmin) / ny

Bxy = 1.
Ly = 2*pi
dx = Ly/ny
dy = dr
```



(a) (b)

Figure 4: The comparsion between simulation results from BOUT++ and the analytical solution at two different positions

```

### Contravariant metric tensor components
g11 = 1.0
g22 = 1.0
g33 = 1.0/Rxy^2

### Covariant metric tensor components
g_11 = 1 / g11
g_22 = 1 / g22
g_33 = 1 / g33
```

Intial conditions and other parameters were identical to the heat conduction problem in the previous section. Results are presented in Fig.(4).

References

[1] BD Dudson, MV Umansky, XQ Xu, PB Snyder, and HR Wilson. Bout++: A framework for parallel plasma fluid simulations. *Computer Physics Communications*, 180(9):1467–1480, 2009.

[2] Jingfei Ma. *The macro-and micro-instabilities in the pedestal region of the Tokamak*. PhD thesis, 2015.

[3] MV Umansky, XQ Xu, B Dudson, and LL Lodestro. Bout code manual. *Science*, 2006.