

# Tokamak grid generator in IDL

B.D.Dudson<sup>†</sup>, M.V.Umansky<sup>§</sup>

<sup>†</sup>Department of Physics, University of York, York YO10 5DD, UK

<sup>§</sup>Lawrence Livermore National Laboratory, CA 94550, USA



June 8, 2012

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Using the Hypnotoad</b>	<b>2</b>
<b>3</b>	<b>Using the grid generator</b>	<b>12</b>
<b>4</b>	<b>DCT</b>	<b>13</b>
<b>5</b>	<b>Finding critical points</b>	<b>13</b>
<b>6</b>	<b>Region numbering</b>	<b>13</b>
<b>7</b>	<b>Separatrices</b>	<b>13</b>

## 1 Introduction

Generation of field-aligned grids is non-trivial for edge tokamak simulations due to magnetic x-points and material surfaces. This grid generator has the following features:

- Written entirely in IDL. Although some might call this a bug, it does mean that nothing needs to be compiled, and (in theory) this code should work everywhere that IDL is installed.
- Automatically adjusts settings when needed. The grid produced can be customised, but the minimum number of inputs is very small (number of grid points and a psi range). The psi range asked for is adjusted to fit within the boundary.
- Can handle an arbitrary number of x-points. Whilst not of obvious benefit since tokamak equilibria are at most double-null, this means the code is quite generalised and can cope with “strange” configurations. Hopefully this will help make the code more resilient.

For a more complicated grid generator which can produce grids conforming to boundaries, see the CARRE grid generator<sup>1</sup> which produces input files for B2, or UEDGE<sup>2</sup> which can also be used to produce input to BOUT/BOUT++.

## 2 Using the Hypnotoad

The easiest way to generate grids is using the GUI called “Hypnotoad” (yes, it is named after the creature from Futurama, for no particularly good reason). Start IDL then run:

```
IDL> hypnotoad
```

This should produce a window which looks like Figure 1. The appearance will vary between operating systems, and the screenshots are from IDL running on Linux.

---

<sup>1</sup>“CARRE: a quasi-orthogonal mesh generator for 2D edge plasma modelling” by R. Marchand and M. Dumberry. *Comp. Phys. Comm.* 96 (1996) pp 232-246

<sup>2</sup>Rognlien et al 1992 *J. Nucl. Mater.* 196198 347

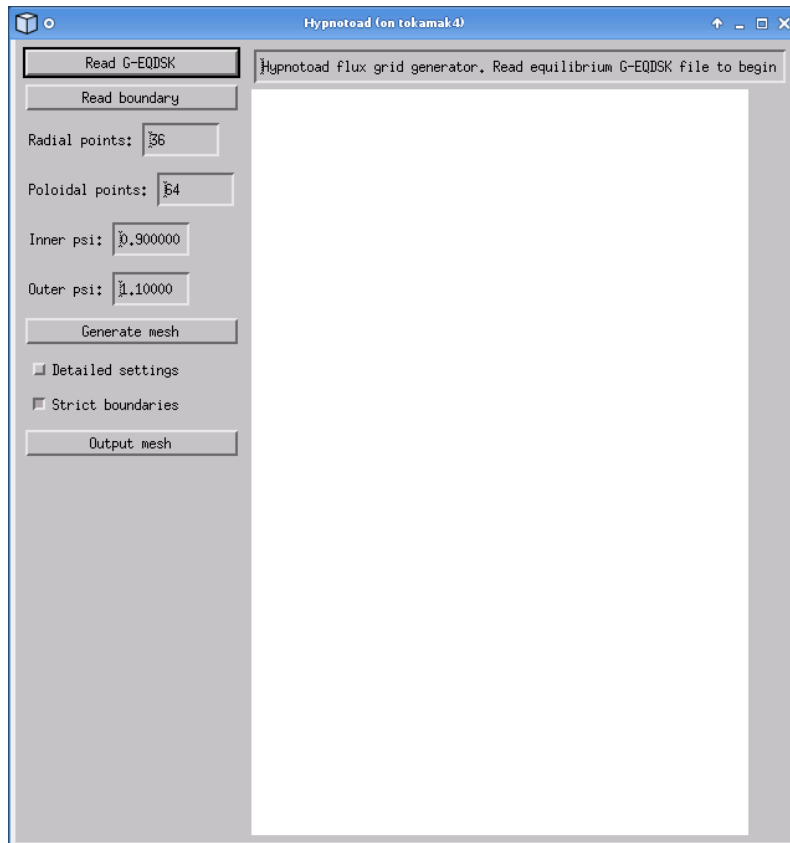


Figure 1: Hypnotoad when started. On the left are buttons to perform processing steps and change some settings, and on the right is the drawing area which will display the grid (white here, but might be black). At the top is a status area which will display text like error messages.

First load the equilibrium file, which must be a G-EQDSK format file (as output by EFIT). Click on the top button “Read G-EQDSK” which will then ask for the file name as shown in Figure 2.

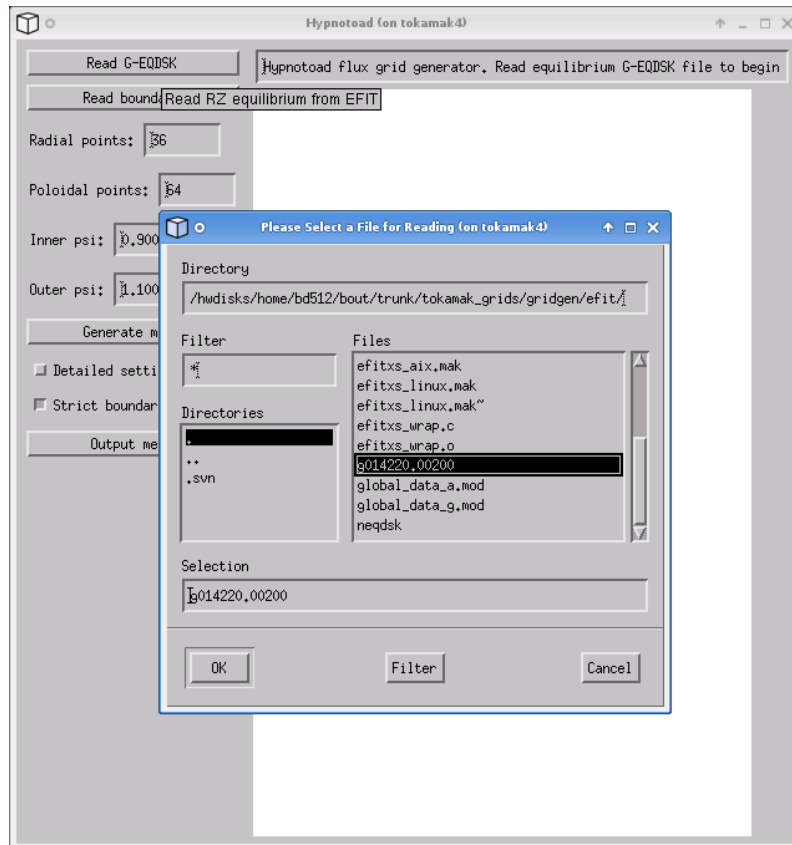


Figure 2: Open a G-EQDSK formatted file. For this example we’ll be using the MAST equilibrium file **g014220.00200** as shown, which is included in the BOUT++ repository (in the **gridgen/efit/** directory)

If the file reading is successful then the grid will be displayed as shown in Figure 3. Flux-surfaces of  $\psi$  are shown in black, but these are just arbitrarily chosen: there is

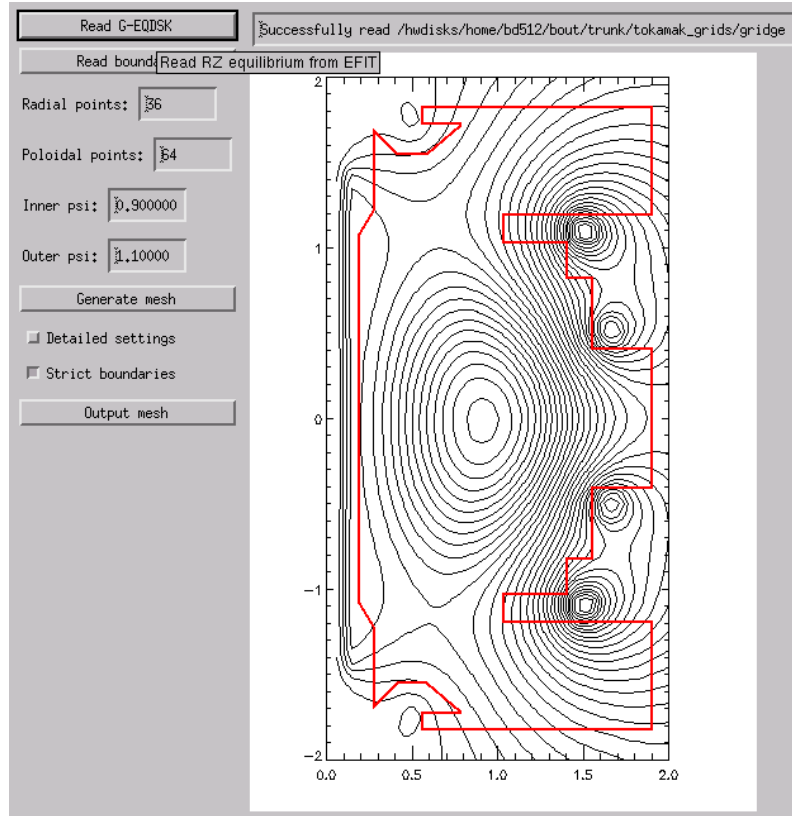
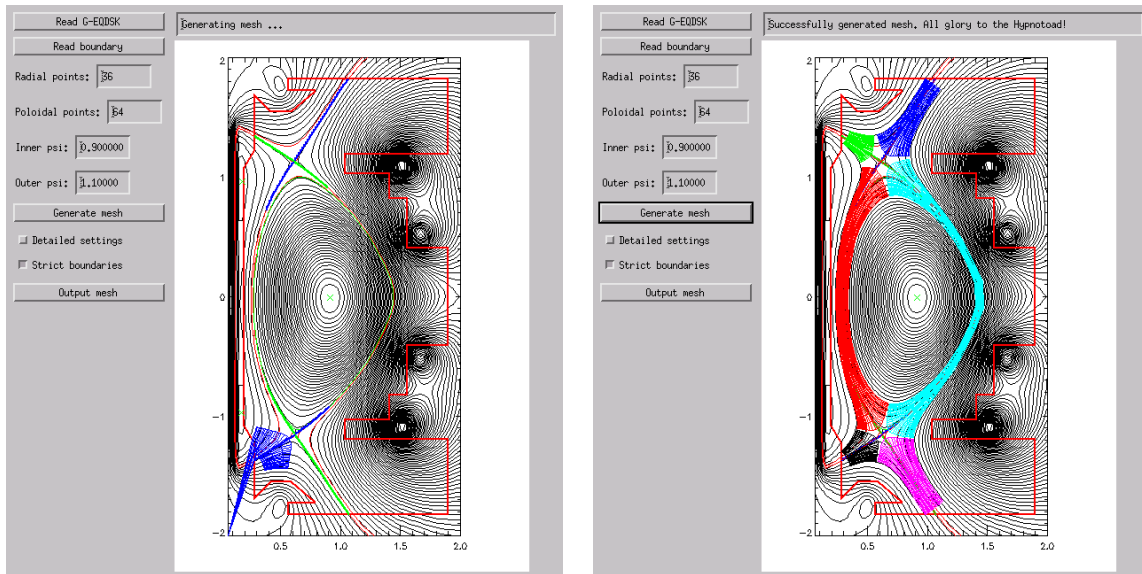


Figure 3: If an equilibrium is read successfully then some psi contours are plotted in black, and if a boundary is defined then this is plotted as a thick red line. This equilibrium has a (slightly) lower double null (LDN) configuration.

no attempt to identify separatrices at this point. In red is the boundary from the G-EQDSK file. Some G-EQDSK files don't specify a boundary, and whilst it's possible to generate grids without one, the result is usually not very good. If this is the case then no boundary will appear, and the status box will contain a warning about missing boundary. To read the boundary from a different G-EQDSK file, click on the second button down labelled "Read boundary".

Once you've got an equilibrium and boundary, enter the number of radial and poloidal points you'd like in the final mesh into the boxes on the left, along with the maximum range of normalised psi (0 = magnetic axis, 1 = innermost separatrix). If this range is too large then it will be reduced automatically.

Clicking on “Generate mesh” runs the grid generator (in **create\_grid.pro**) the detailed of which are given in more detail in the following chapters. Initially the x-points are located, and those which are inside the boundary and the range of psi specified are selected. These are then refined and analysed to determine the directions corresponding to the divertor legs and the core separatrixes. Constant psi surfaces are followed to the target plates. Figure 4(a) shows these lines and the initial attempt at meshing the lower left divertor leg. Note that this meshing won’t always start at the lower left: Meshing starts from the innermost x-point (i.e. the one with normalised  $\psi = 1$ ) and proceeds clockwise around the domain. For an Upper Double Null (UDN) configuration the meshing therefore starts at the top right (outer upper divertor leg). You can adjust the settings



(a) Initial steps in generating the grid. X-points are found and the legs traced to the boundary. Regions are meshed and intersections with the boundary found (where blue lines are drawn to the origin)

(b) Finished mesh with regions colored (black=1, red=2, green=3, blue=4, turquoise=5, purple=6)

Figure 4: Generating a mesh

and re-generate the mesh if needed. The switch called “Strict boundaries” determines whether grid points are allowed to cross the boundary. If this is switched off, the divertor legs will still end at the boundary, but the radial mesh is allowed to cross the boundary.

Once you're happy with the mesh, click on "Output mesh" to create a grid file for input to BOUT/BOUT++. This will prompt for a file to write, and if you're overwriting a file it will ask you to confirm.

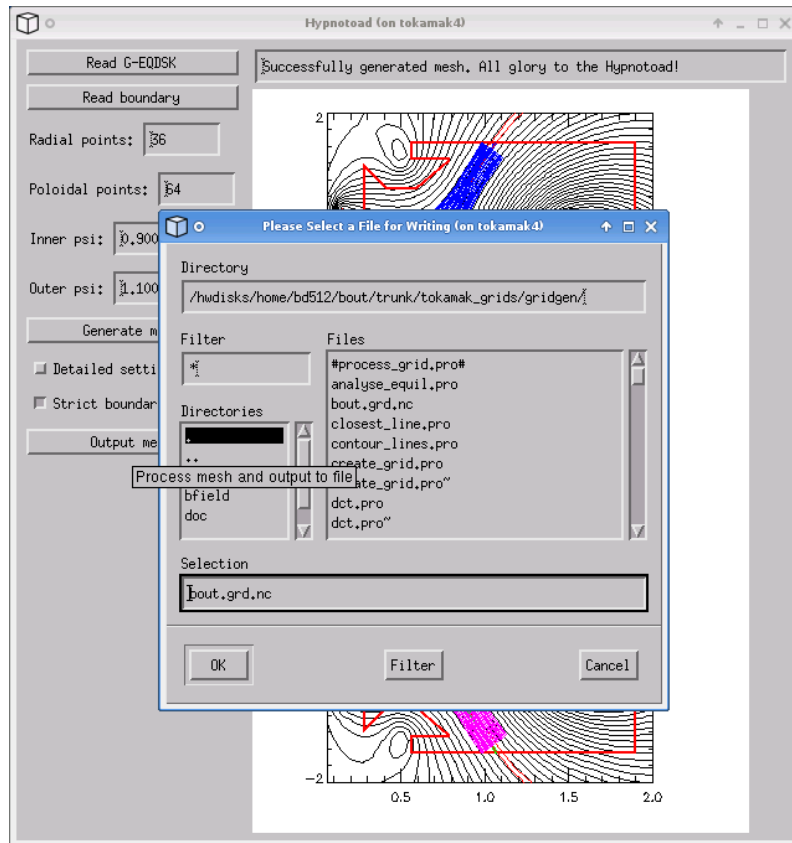


Figure 5: Choosing name of output grid file

Whereas mesh generation is automatic, turning a mesh into something which can be used for simulations usually requires some user intervention. This is mainly about checking that the equilibrium is reasonably good and that the pressure and current profiles look sensible. If these are not good then the simulation could have numerical problems, or just produce results which are not correct.

Firstly the pressure gradient is calculated using

$$\mu_0 h_\theta \frac{\partial P}{\partial x} = -B_\theta \frac{\partial}{\partial x} (B_\theta h_\theta) - B_\zeta h_\theta \frac{\partial B_\zeta}{\partial x} - \frac{B_\zeta^2 h_\theta}{R} \frac{\partial R}{\partial x}$$

(see ??) for derivation.  $h_\theta$  is calculated by measuring the geometric distance between points,  $B_\zeta$  from the  $f(\psi)$  profile, and  $B_\theta$  from derivatives of  $\psi$  (done in **create\_mesh.pro** using DCTs). This calculated pressure is compared to the pressure profile from the G-EQDSK file as shown in Figure 6.

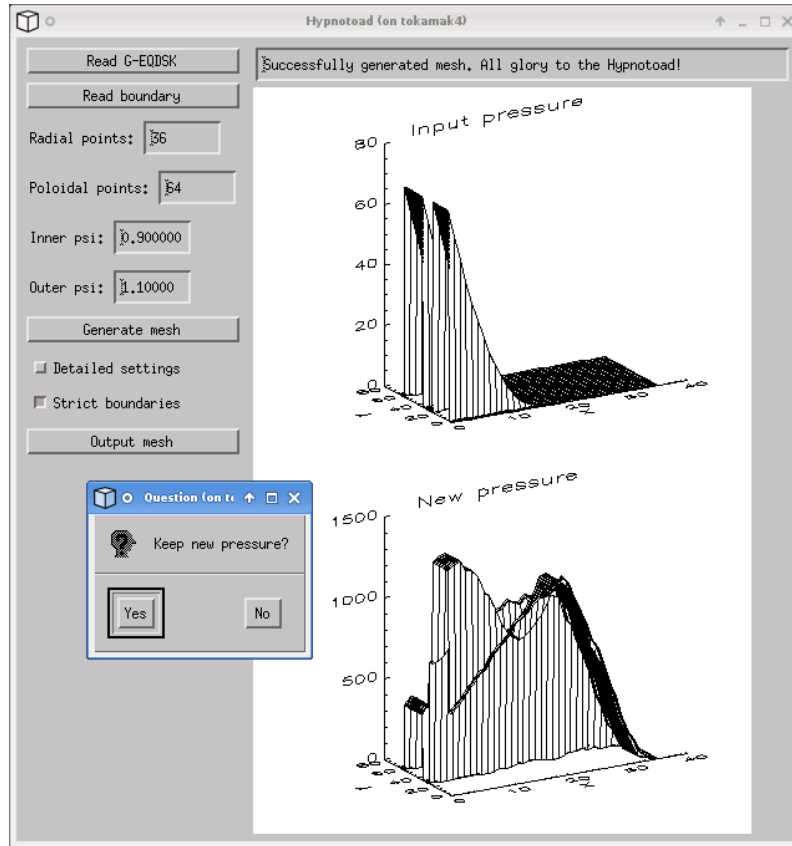
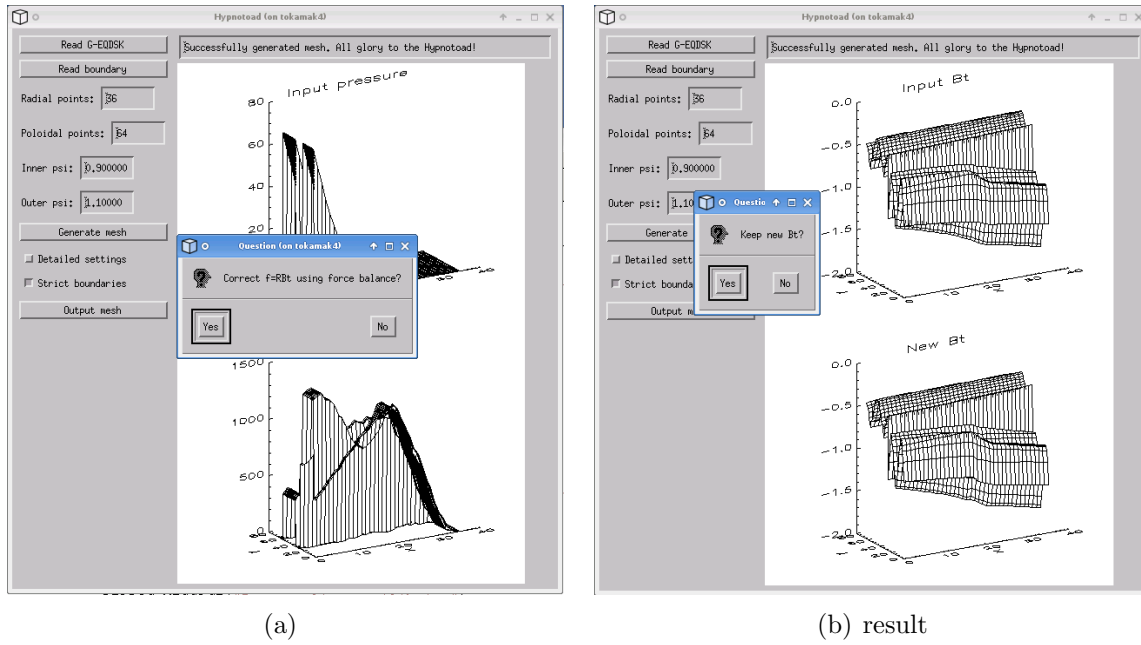


Figure 6: Calculating pressure from force balance



Figure 7: Calculating  $f = RB_\zeta$  from force balance

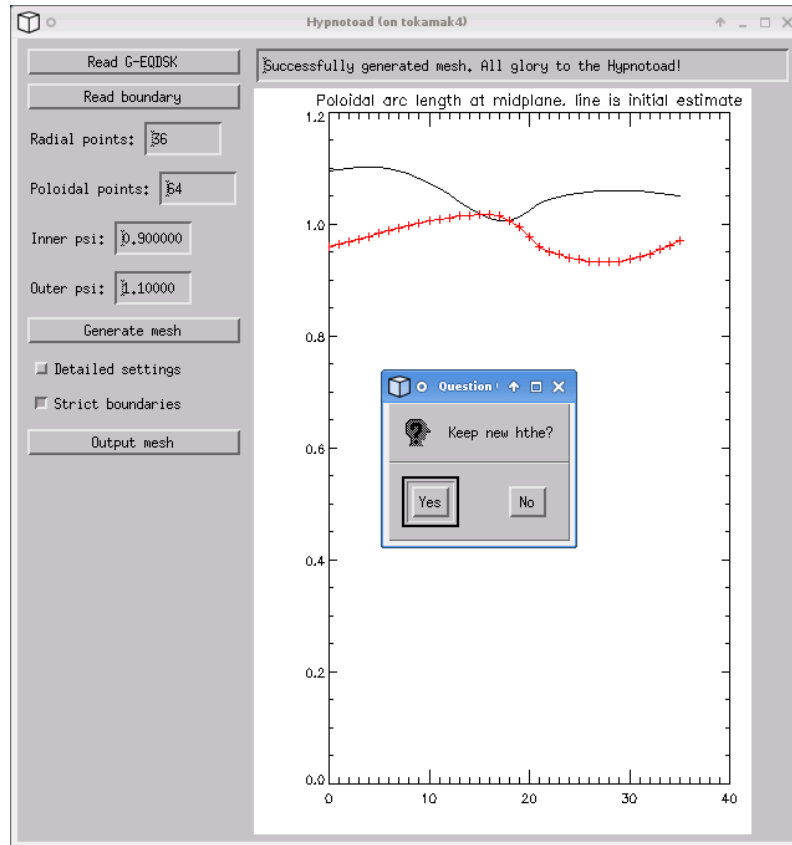


Figure 8: Calculating poloidal arc length  $h_\theta$  from force balance

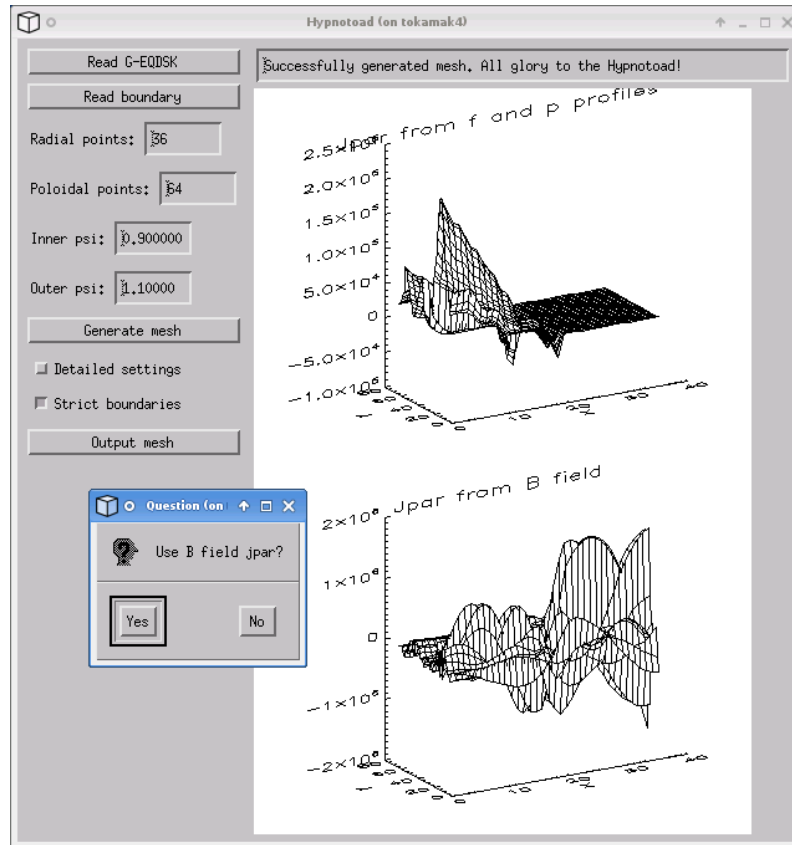


Figure 9: Calculating  $J_{\parallel}$  from  $f(\psi)$  and  $p(\psi)$  (top), and from  $\mathbf{B}$  field (bottom)

```

Generating plasma profiles:
  1. Flat temperature profile
  2. Flat density profile
  3. Te proportional to density
% Compiled module: GET_INTEGER.
Profile option:1
Setting flat temperature profile
% Compiled module: GET_FLOAT.
Temperature (eV):10
Maximum density (10^20 m^-3):      0.216518
Is this ok?y
Setting rmag =      1.46673
Setting bmag =      1.79263
Writing grid to file /hwdisks/home/bd512/bout/trunk/tokamak_grids/gridgen/bout.grd.nc
% Compiled module: FILE_OPEN.
% Compiled module: NCDF_EXISTS.
% Compiled module: FILE_WRITE.
% Compiled module: REVERSE_INDS.
% Compiled module: FILE_CLOSE.
DONE

```

### 3 Using the grid generator

The function `create_grid` takes a 2D array of  $\psi$  values, and produces an orthogonal mesh aligned with the flux-surfaces.

Settings to control the resulting mesh are:

- **psi\_inner**, the normalised  $\psi$  of the innermost flux surface. This can be either a scalar or an array:
  - **scalar**: This value is used for the core and all PF regions
  - **array[0]**: The inner normalised  $\psi$  for the core
  - **array[1..n\_xpoint]**: Inner  $\psi$  to use for each PF region (see section 6)
- **psi\_outer**, normalised  $\psi$  of outermost surface. Can also be either a scalar or array:
  - **scalar**: This value is used for the core and all PF regions
  - **array[0..(n\_xpoint-1)]**: Outer normalised  $\psi$  for each SOL region (one per x-point)
- **nrad** Number of radial grid points

- **scalar**: Total number of radial grid points. Automatically divides this between regions.
- **array[0]**: Number of radial grid points in the core
- **array[1..(n\_xpoint-1)]**: Radial grid points between separatrices (going outwards from core to edge)
- **array[n\_xpoint]**: Radial grid points outside last separatrix
- **npol** Number of poloidal grid points.
  - **scalar**: Total number of points. Divides between regions based on poloidal arc lengths
  - **array[0..(3\*n\_xpoint-1)]**: Number of points in each poloidal region. See section 6 for numbering scheme.
- **rad\_peaking**
- **pol\_peaking**

## 4 DCT

DCT of 2D NxM  $f(x, y)$

$$F(u, v) = \sqrt{\frac{2}{N}} \sqrt{\frac{2}{M}} \Lambda(u) \Lambda(v) \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(i, j) \cos \left[ \frac{\pi u}{2N} (2i + 1) \right] \cos \left[ \frac{\pi v}{2M} (2j + 1) \right]$$

where  $\Lambda(i) = 1/\sqrt{2}$  for  $i = 0$ , and  $\Lambda(i) = 1$  otherwise

## 5 Finding critical points

To find x- and o-points,

## 6 Region numbering

## 7 Separatrices

Having found the x-point locations, the separatrices need to be found. First step is to calculate the lines going through the x-point:

Close to an x-point, approximate the change in  $\psi$  by

$$\delta\psi = \frac{1}{2}\psi_{xx}x^2 + \frac{1}{2}\psi_{yy}y^2 + \psi_{xy}xy$$

The two lines through the x-point are then given by where this is zero:

$$\frac{1}{2}\psi_{yy}y^2 + \psi_{xy}xy + \frac{1}{2}\psi_{xx}x^2 = 0$$

Which has the solution

$$y = \frac{-\psi_{xy}x \pm \sqrt{\psi_{xy}^2x^2 - \psi_{yy}\psi_{xx}x^2}}{\psi_{yy}}$$

i.e.

$$y = \frac{1}{\psi_{yy}} \left( -\psi_{xy} \pm \sqrt{\psi_{xy}^2 - \psi_{yy}\psi_{xx}} \right) x$$

Note that if  $\psi_{yy} = 0$  then the solutions are  $x = 0$  and  $y = -\frac{\psi_{xx}}{2\psi_{xy}}x$