Generating a CWS, Running NESCOIL, and Cutting Coils from Current Potential

Last modified Wednesday, November 20, 2013

(1) Compile Bnorm with Satisfactory Parameter Values:

I work with a standard STELLOPT directory structure. Executable **xbnorm** is in **/p/pies/Pomphrey/STELLOPT_CODES/STELLOPT/My_stellopt/BNORM/Vrelease**. Within that Vrelease area I have several versions of **xbnorm** compiled with different parameter values. The executables are of the form **xbnorm_nu_nv_mf_nf_md_nd** (eg **xbnorm_64_64_10_10_20_20**).

In NESCOIL, parameters **md** and **nd** define the max poloidal and toroidal modenumbers for the Fourier description of the CWS. Parameters **mf** and **nf** set the max poloidal and toroidal modenumbers for the current potential Fourier coefficients. Rather unintuitively, although potential coefficients are calculated within NESCOIL, the **mf, nf** parameters are set in **bfield_n.f** which is in the **BNORM/vsource** directory! Typically, we would want to have a NESCOIL solution (current potential) that reproduces the Bn on the LCFS to within some accuracy (Berr < Tol ... Eg Max error < 5% and average error < 1% say (see LPKU “Modular Coils and Plasma Configurations for QA Stellarators”, Nuc Fus **50 (2010)** 125005**,** p4 of 15). **Typical parameter values for scoping studies are nu=64, nv=64, mf=10, nf=10, md=20,nd=20**, but be prepared to do a convergence study particularly on **mf, nf** to get satisfactory reproduction of the target equilibrium.

(2) Run Bnorm with some VMEC equilbrium wout.extension:

Execute with syntax **./xbnorm extension 0.2**.

This version of xbnorm has a three-fold purpose: First, it generates file **bnorm.extension** which is a Fourier decomposition of B.n on the plasma LCFS given as a list of m, n, and Bmn values (with NESCOIL convention - ie., corresponding to the angle combination mu + nv as opposed to VMEC’s mu - nv). Second, it generates a coil winding surface (CWS) with a uniform offset to the LCFS. In the case of **./xbnorm extension 0.2**, the offset is 0.2m and is uniform over the entire toroidal surface. The code assumes a max poloidal and toroidal modenumber for the CWS of **md** and **nd**, which are set in **bfield_n.f**. The next section describes how to generate a CWS with non-uniform offset. Third, a NESCOIL input file is created named **nescin.extension** which communicates essential equilibrium parameters to NESCOIL, shapes of the plasma LCFS and coil winding surface CWS, and sets default values of coil topologies.

One should ALWAYS check that the CWS makes sense (see bottom of next section).

(3) Generate a CWS with non-uniform offset from the LCFS:

Skip this section if a uniform offset CWS is OK!

In **dir2** = **dir1/NESCOIL/GENSURF** where **dir1** = **/p/pies/Pomphrey/CONSULTING/LPKU_A6_CONFIG_FROM_NUC.FUS.2010**, is **gensurf_develop.f**, a simple modification of LPKU's original **/u/ftp/pub/lpku/STELLOPT_CODES/NESC/gensurf.f** code. The modified code allows a simple toroidal variation in LCFS to CWS distance, Δ, as well as the poloidal variation allowed in LPKU's version.
As before, an input file **bndry.extension** is expected by the CWS generator. An example of **bndry.extension** is **dir2/bndry.lpku_A6_lowiota**. Line 1 specifies the number, **nmn**, of non-zero Fourier cpts of the plasma boundary and the number of field periods. The next **nmn** lines list the m, n, Rmn and Zmn values specifying the boundary. Following this is an integer, **ndel,** specifying the number of u values at which Δ(u,v) is constrained. Whereas before the **ndel** values in the 1st column that follow are the Δ(u) values at all toroidal angles, now the entries in the 1st column are distance values at the v=0 symmetry plane while the 2nd column are distance values at v=0.5. A simple interpolation scheme (SMOOTHSTEP Hermite Interpolation) interpolates Δ(u) as a function of v between v=0.0 and v=0.5. For example,

5
0.0 3.3 1.4
0.3 1.4 1.4
0.5 1.4 1.4
0.7 1.4 1.4
1.0 3.3 1.4

presents a scheme where the outboard Δ is 3.3m at u=0, v=0 (allowing for neutral beams) decreasing to Δ=1.4m on the inboard side at v=0, while at v=0.5 there is a uniform separation of Δ=1.4m for all u values. The selection of u values is best done by trial and error and visual inspection since u is not geometric! For this example and the particular (ARIES) equilibrium, u = 0.3 is near the tip of the bean.
Script file **comp** can be used to generate an executable by typing
./comp gensurf_develop ➜ gensurf_develp.x
which can be executed by typing
./gensurf_develop.x bndry.extension ➜ nesin.extension that contains both the plasma LCFS and CWS.

NOTES:

- *ALWAYS inspect the* **outcurve.extension** *file that is produced to check that the DESCUR residuals are reasonably small, and*
- *ALWAYS plot a superposition of the CWS and LCFS surfaces for a reality check! To do this, compile and run* **dir2/plot_nesin_surfaces.f** (using **comp**) *and run the executable using* **./plot_nesin_surfaces.x nesin.extension**. *This produces output files* **nesin.extension_plas.pts** *and* **nesin.extension_cws.pts** *which are the boundary points at toroidal cross sections v=0, 0.25, and 0.5. These can be plotted using GNUPLOT.*
- *Parameter values for the generation of the CWS appear at the top of* **gensurf_develop.f**

(4) Run NESCOIL:

Executable **xnescoil** is in
/p/pies/Pomphrey/STELLOPT_CODES/STELLOPT/My_stellopt/NESCOIL/Vrelease.

Syntax for execution is **./xnescoil nescin.extension ➔ nescout.extension**

Please note that required input file is **nescin.extension**, the version produce by xbnorm**,** not **nesin.extension** described in (3) above. If a non-uniform offset CWS has been generated, extract the Fourier description of the CWS from **nesin.extension** and paste into the **nescin.extension** file before executing **xnescoil**.

Before running NESCOIL, check that the coil topology (modular, saddle, wavy pf) has been specified correctly - the **cut, cup** values. (Secular terms in the current potential **Φ(u,v)** are **cut*u + cuv*v**). The default values are **cut=0, cup=1, ibex=0** corresponding to modular coils. For saddle coils, use **cut=0, cup=0, ibex=1** and for wavypf use **cut=1, cup=0, ibex=1**

NESCOIL calculates current potential for unit poloidal current. The **curpol** value output in **nescin.extension** will be used (and explained) later to get physical units of current.

Output file **nescout.extension** has two parts: The first part is a replication of **nescin.extension**. The essential output is the calculated current potential denoted **Phi(m,n)**, the error in matching B.n at the plasma edge (Berr ave, max, var), and the surface current diagnostics giving the complexity, etc.

Miscellaneous notes:

The **curpol** *value in* **nescin.extension** d*iffers from the value of* **curpol** *from th*e **xbnorm** *formed from LPKU's* **nesc.zip** *version by a factor of 4π*1.e-7. Unfortunately, there are a bunch of similar codes floating around so be careful which version of the NESCOIL suite you are dealing with!*

The value of **phiedge** *in* **nescin.extension** *is the VMEC* **phiedge** *divided by 2π.*

(5) Process and Plot the Contours of the Current Potential:

To obtain a contour plot of the current potential (when we cut coils they will lie parallel to contours) first create a file called **pot.coeffs.extension** which will contain the Fourier components of the potential, Φ(m,n).
The first 2 lines should be the **cut** and **cup** values from nescout.extension, e.g.,
cut 0.0
cuv 1.0

and the remaining lines are the list of m, n, Φ(m,n) values copied from **nescout.extension**.
Extract source **cp.f** from LPKU's **/u/ftp/pub/lpku/STELLOPT_CODES/NESC/nesc.zip** and **unzip nesc.zip Other_Codes/cp.f** and form executable **cp.x** by typing **./comp cp**. (Alternatively, just copy my **cp.x** from **/p/pies/Pomphrey/CONSULTING/LPKU_A6_CONFIG_FROM_NUC.FUS.2010/NESCOIL**).

Next, copy the idl routine and input file that will be used to plot the contours of Φ(u,v):
cp /u/ftp/pub/lpku/STELLOPT_CODES/NESC/sample_input.zip .
unzip sample_input.zip cp.pro cp.inp
(Alternatively, copy **cp.pro** and **cp.inp** from **p/pies/Pomphrey/CONSULTING/LPKU_A6_CONFIG_FROM_NUC.FUS.2010/NESCOIL**).

(a) Run **cp.x** using syntax **./cp.x < pot.coeffs.extension > uvfile**
(b) Note the number of lines in output file **uvfile** by typing UNIX command **wc -l uvfile**
(c) Copy **uvfile** into **uv.dat**, insert the integer value of the number of lines as the first line of **uv.dat**, and close the file
(d) Run **idl** using **cp.pro**
idl
cp
exit
choosing an appropriate grid size for the contouring and number of contours. These are specified in **cp.inp**. A typical **cp.inp** can be found in **/p/pies/Pomphrey/CONSULTING/LPKU_A6_CONFIG_FROM_NUC.FUS.2010/NESCOIL**.
The contour plot will appear on the screen and, on exit, a postscript file named **idl.ps** will be created. Rename this to some sensibly identifiable **idl.name.ps**.

(6) Cut coils from the Current Potential:

The next steps convert current potential contours to coil filaments. The relevant source files can be found in LPKU's **nesc.zip**, and compiled locally, or copy executables from my **/p/pies/Pomphrey/CONSULTING/LPKU_A6_CONFIG_FROM_NUC.FUS.2010/NESCOIL**.

unzip nesc.zip Other_Codes/uv2xyz.f Other_Codes/rwcoils.f
and form executables **uv2xyz.x** and **rwcoils.x** by typing
./comp uv2xyz
./comp rwcoils

uv2xyz.x requires an input file specifying the CWS. In these instructions, let's call this file **cws** but it should really have a sensible identifier. This file can be obtained from **nescin.extension** by copying all lines after
Table of fourier coefficients

m n cr2(m,n) cz2(m,n)

(a) Execute **uv2xyz.x** using syntax **./uv2xyz.x cws uvfile 3 junk fd.extension**. Note the number of field periods (3) is input. The two files **junk** and **fd.junk** are intermediate output files on the way to producing a standard **coilsfile** for use by free-boundary VMEC. Columns 5 and 6 of **fd.extension** are the θ/2π, nfp*ϕ/2π values for the coils on the surface over the full torus. The currents in column 4 of the filament file **fd.extension** incorporate the unit poloidal current normalization of NESCOIL. Some more processing is necessary before these files are really useful:

(b) Delete columns 4 and 5 of **fd.extension** and assign a single group number and name to each coil. A simple way of doing this is to use part of LPKU's **cutcoils** script (the part that uses some tedi editor commands):

tedi fd.extension
nv
dac1,],60
dop1,],40,60;0.000;ac; 1 Modular;ndo
end

(c) To produce a coils file that can reproduce the VMEC equilibrium, we need to assign the correct coil currents. This is accomplished by executing **rwcoils.x**, using namelist input file **indata** that can be extracted from **/u/ftp/pub/lpku/STELLOPT_CODES/NESC/sample_input.zip** or simply copied from **/p/pies/Pomphrey/CONSULTING/LPKU_A6_CONFIG_FROM_NUC.FUS.2010/ NESCOIL**. Here is an example:

&indata
coils_in="fd.extension"
coils_out="coils.name"
lscalecur=T
scalecur=97826065.2765197
/

The value of **scalecur** in **indata** should be set to the value of **curpol** output from **xbnorm** in the **nescin** file divided by 4π*1.e-7.

To execute **rwcoils.x** use syntax **./rwcoils.x.** This will produce a standard coils file called coils.name which should reproduce the desired equilibrium if the Berr residual is small enough (see section (4) above). The group numbers and coil names need to be modified by hand.

(d) Now we can generate an **mgrid** file and run VMEC free-boundary, comparing the physics of the free-boundary run with that of the original fixed boundary equilibrium. **EXTCUR** value should be **curpol**/(4π*1.e-7)*number in column 4 of **fd.junk** [1./(#contrs*nfp)]

Summary of essential steps

./xbnorm_64_64_6_6_20_20 lpku_A6_lowiota
--> bnorm.lpku_A6_lowiota, nescin.lpku_A6_lowiota
Then substitute the desired cws (ecws_from_tbs2-mc_wc-5_curves.txt) into
nescin.lpku_A6_lowiota

./xnescoil nescin.lpku_A6_lowiota
--> nescout.lpku_A6_lowiota

Copy and paste the Phi(m,n) from nescout.lpku_A6_lowiota into file pot.coeffs, makink sure to paste
cut 0.0
cuv 1.0
as a header in pot.coeffs

Create input file cp.inp with contouring instructions. Here is an example:
Current Potential test$
$
toroidal - v$
poloidal - u$
0, 0, 128, 128, -6, 0 / cpol, ctor, nu, nv, nc1, nt1
0.

Run the contouring code cp.x:
./cp.x < pot.coeffs > uvfile
--> uvfile

Run uv2xyz.x:
./uv2xyz.x cws uvfile 3 junk fd.lpku_A6_lowiota
--> fd.lpku_A6_lowiota

Remove the 3-line header and 1-line footer from fd.lpku_A6_lowiota

Run splinefit with the desired number (n) of control points
./splinefit fd.lpku_A6_lowiota -fp 3 -n 16 -c cws
--> fd.spline, forstellopt (fd.spline is the file needed by COILOPT++ and forstellopt has the spline information that can be pasted into the appropriate place in the STELLCOPT input file)

The next two steps allow you to view the coils on the u-v plane to compare the splinefit with the
u-vcoils that were cut from the current potential:

Form a VMEC/mgrid coilsfile
/u/jbreslau/bin/spline2coils fd.spline > coils.lpku_A6_lowiota
--> coils.lpku_A6_lowiota

Form a fd.trunc file from the coils file
/u/jbreslau/bin/coils2trunc coils.lpku_A6_lowiota cws 3 -o fd.trunc
--> fd.trunc