

STELLOPT TUTORIAL

Neil pomphrey July 2020

These notes can be considered a tutorial on how to run STELLCOPT - the **STELL**arator **Coil** **OPT**imization code.

Introduction

Traditionally, our coil design approach has been a staged approach:

1. STELLCOPT is run in fixed-boundary mode (LFREEB = F) varying the Fourier coefficients of the plasma boundary (RBC, ZBS). A plasma is designed whose major radius ($\langle R \rangle$), aspect ratio ($A = \langle R \rangle / \langle a \rangle$) and toroidal magnetic field (linked toroidal flux, PHIEDGE) have desired values, and whose physics properties (transport and stability) at some target β are attractive.
2. The associated normal component of magnetic field at the plasma boundary, B_n^{plas} , becomes a target for a NESCOIL or REGCOIL calculation. Here, a current potential Φ on a specified **Coil Winding Surface** (CWS) is sought which produces an external magnetic field whose normal component at the plasma boundary, B_n^{ext} , cancels the plasma component. The CWS, parameterized by poloidal and toroidal angles u and v , typically has a non-uniform distance offset from the plasma. The offset distance can be large for reactor relevant calculations.
3. The current potential $\Phi(u, v)$ is discretized using a contouring algorithm and a limited set of contours (eg 3 in each half period for the 3-field period stellarator symmetric NCSX configuration) are selected as discrete coils (uv -coils) on the u, v plane, which are trivially mapped to cartesian coordinates from which a coilsfile for VMEC calculations is built.

A description of the steps and codes used to execute steps 1-3 above can be found in document "**Generating a CWS, Running NESCOIL, and Cutting Coils from Current Potential**".

Step 2 of the traditional approach minimizes $|B_n^{ext} - B_n^{plas}|$ and produces a residual δB_n fitting error. Step 3 boosts this error. Using a realistic CWS can give rise to coils with δB_n (rms and max) large enough that when free boundary VMEC equilibria are calculated using the discrete coil set, the resulting plasma can be substantially different from, and inferior in performance to, the original design target. Indeed, the residual fitting error can result in VMEC failing to converge!

Some of the drawbacks of the traditional coil design approach were recognized in the early 2000's the ORNL design group. Quoting from Strickler, Berry, Hirshman "**Designing Coils for Compact Stellarators**", *Fusion Science and Technology*, **107 - 115 (2002)** :

- *"A method is presented for designing coils for compact stellarators. In contrast to methods that select a finite number of coils from an optimal continuous surface current distribution, the COILOPT code solves for the optimal parameters in an explicit representation of modular coils on a toroidal winding surface that is well separated from the plasma boundary, together with the coefficients of the winding surface. The problem is posed as a balance between approximating a prescribed magnetic configuration and satisfying certain critical engineering requirements ..."*

This COILOPT approach eliminates Step 3 of the traditional approach. Indeed, COILOPT was integrated into the ORNL version of STELLOPT by Strickler et al and presented at the 2002 IAEA meeting so that even step 2 was eliminated. Here is an extract from their IAEA abstract:

- *"The integration of the COILOPT model, based on explicit representations for modular coils and coil geometry constraints, into the stellarator optimization package STELLOPT, provides a unique and important computational tool for the design of compact stellarators. This self-consistent analysis ensures that physics and engineering criteria are simultaneously satisfied ..."*

Subsequently there was a divergence of the ORNL and PPPL versions of STELLOPT and very little exercising of this COILOPT-within-STELLOPT approach at PPPL.

In 2012 Josh Breslau began construction of a local version of COILOPT, written in the C++ programming language and called COILOPT++. Working with a cubic B-spline representation of Modular, Saddle, or Poloidal Field coils on one, or several, fixed coil winding surfaces the code incorporates a suite of engineering constraints such as coil curvature, torsion, coil-to-coil spacing. Stay-out zones and zones can be introduced, and true straightening of MC coils over defined distances. The stay-out zones have been used to define ports that allow access to blankets and internal components.

More recently the COILOPT++ capabilities have been incorporated within STELLOPT to form STELLCOPT.

Coil Design with STELLCOPT

When STELLCOPT operates in free-boundary mode (LFREEB = T) - for example for the flexibility and robustness studies in support of the NCSX PDR - a coilset (and therefore a coilsfile) is implicitly assumed. An mgrid file corresponding to that coilsfile is specified in the input file, as are initial coil currents in the EXTCUR array. STELLCOPT also requires an mgrid file and EXTCUR coil current values **as well as** an explicit specification of the knot values and control points corresponding to the coils. In general, a splinefit cannot be perfect. So it should be clear that what is expected for consistency of initial conditions for the run is that splines --> coils --> mgrid file.

Let's assume a fixed boundary target equilibrium has been determined, a suitable coil winding surface has been defined, NESCOIL or REGCOIL has been run, and a coilsfile generated corresponding to the chosen contours of the current potential. To generate a spline representation of the coils, two utility codes (**coils2trunc** and **splinefit2**) must be run using the following syntax (for a 3-field period device):

1. `/u/jbreslau/bin/coils2trunc coils.ext cws 3 -o fd.ext`
2. `/u/jbreslau/bin/splinefit2 fd.ext -fp 3 -n 16 -c cws`

The value of n in the splinefit call corresponds to the number of control points to be used in the spline fit. Since we are using cubic B-splines, the number of knots equals the number of control points + 4.

Two files are output from the **splinefit2** code, namely **forstellopt** and **fd.spline**. The first of these (**forstellopt**) contains the knots and control points of the spline fit to the coils in the uv plane. The second (**fd.spline**) is irrelevant for STELLCOPT but is a required file for a COILOPT++ run. The information it contains is equivalent to that in forstellopt but the format is different.

The following plot shows an overlay of coils on the u-v plane of a CWS comparing different numbers of control points (10 - green, 16 - red, and 32 - blue) used in the spline fit.

n = 10 gives a rather crude approximation to the NESCOIL curves

n = 16 recovers the essential features of NESCOIL and is a satisfactory starting point for STELLCOPT runs.

n = 32 essentially overlays the NESCOIL solution.

All we really need to start STELLCOPT is an initial spline representation whose corresponding coilsfile will construct a free boundary equilibrium. If that equilibrium has good physics properties then great ... but this is not necessary! STELLCOPT will vary the control point locations to produce a coilset with improved physics. Later, we can increase the number of knots / control points (degrees of freedom) to further improve the physics if needed.

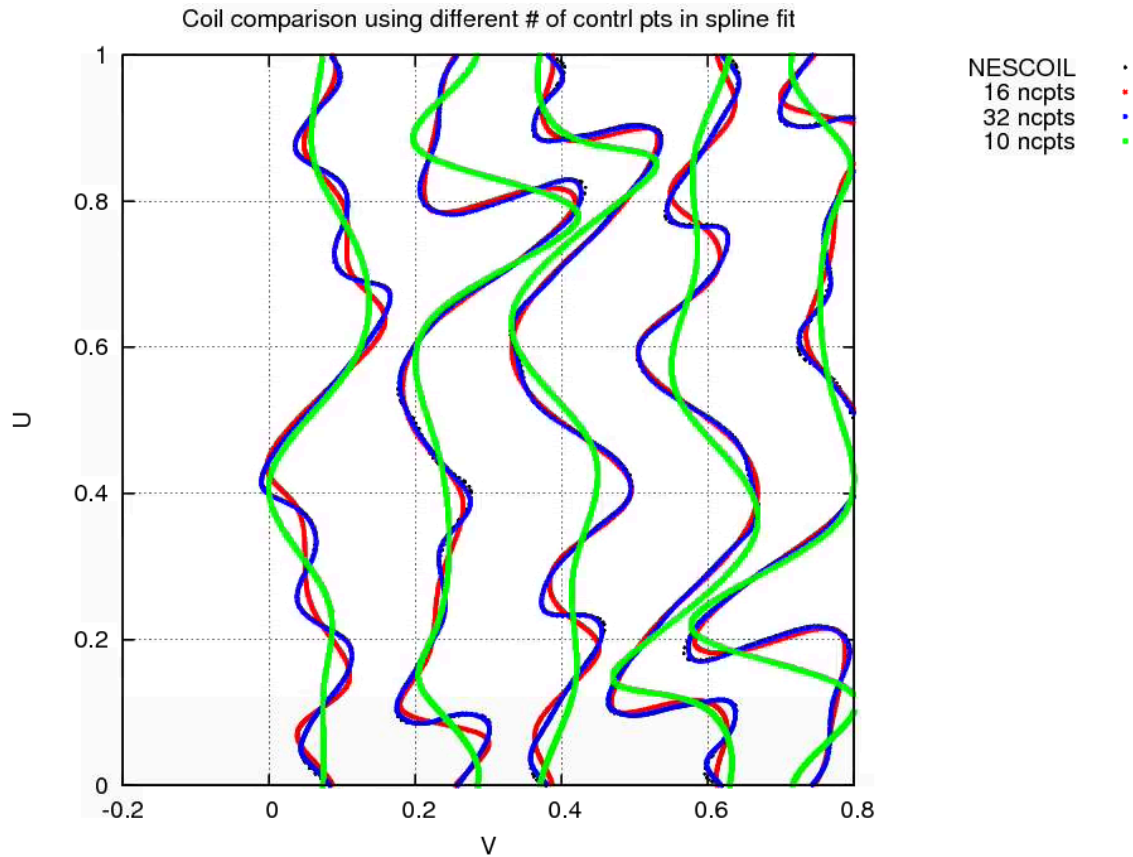


Fig 1

Let's have a look at the plasma boundary shapes of the target plasma `lpku_A6_lowiota` configuration (the fixed boundary target which provides the B_n for the NESCOIL coil design), the free-boundary reconstruction using the NESCOIL solution, and the free-boundary reconstructions from the 10 and 16 cntrl pt coils.

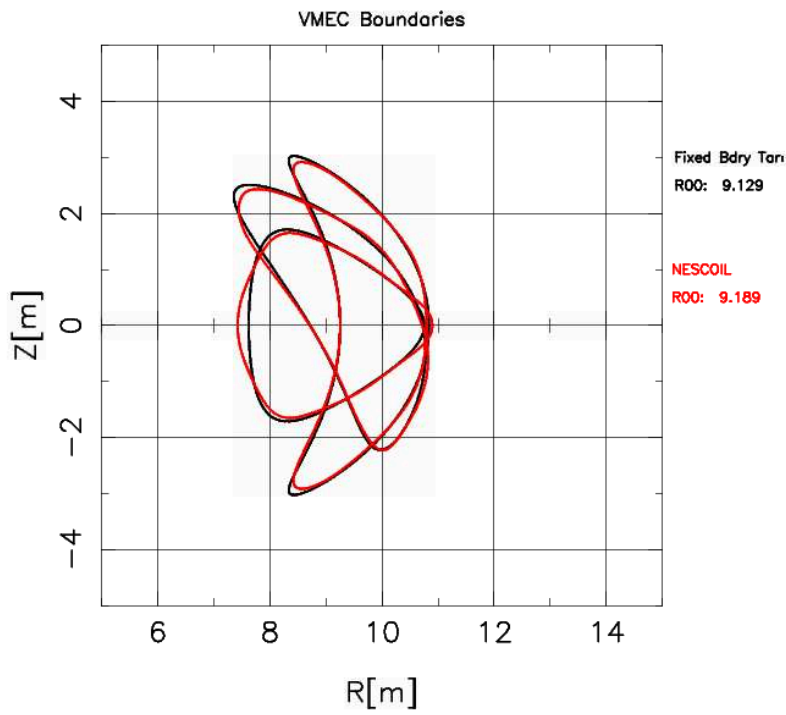
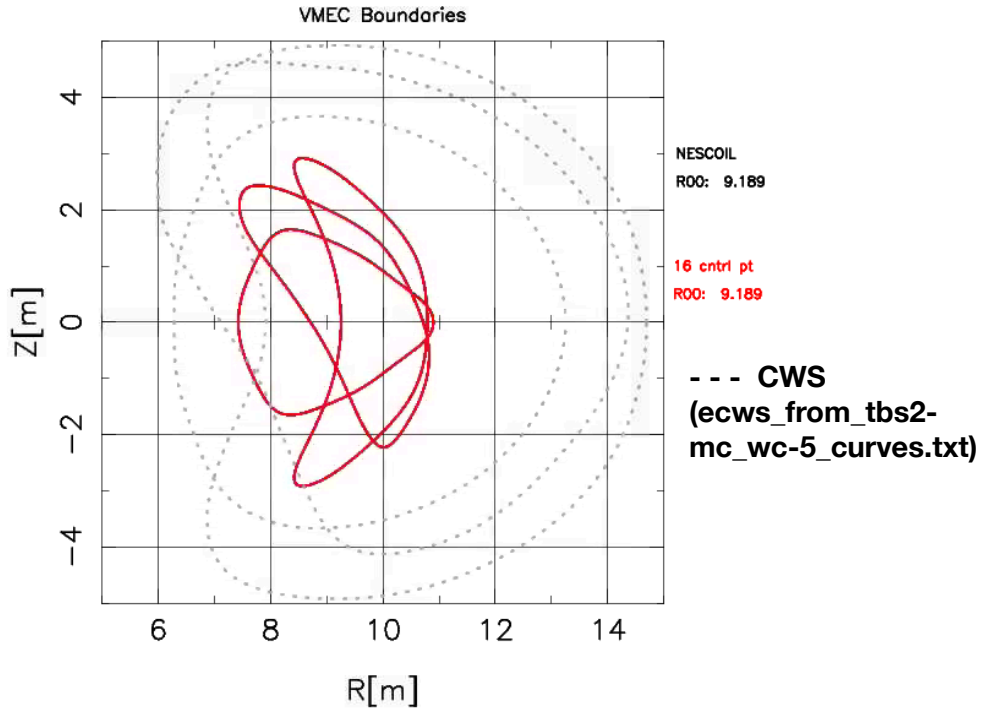


Fig 2a, b

The 10 cntrl pt equilibrium did not converge so we can't show the plot.

The 16 cntrl pt and NESCOIL free boundary VMEC solutions (Fig 2a) completely overlay one another, which is interesting since the coils do not overlay in Fig 1. --- but there is little difference in uv-coil shape on the inboard side.

The NESCOIL VMEC boundary differs substantially from the target equilibrium on the inboard side of the $v=0.5$ cross section and in the tip region at $v=0$. So we can expect that the reconstructed plasma from the NESCOIL and splined solutions have different physics properties from the NESCOIL target. It turns out that the $\epsilon_{eff}^{3/2}$ values for the 16 cntrl point solution are not horrible ($6.317e-4$, $8.753e-4$ and $1.414e-3$) at the ($j_s = 13/49$, $25/49$ and $37/49$) surfaces. However, the kink stability is seriously degraded by the change in shape ... the kink eigenvalues for the $n=0$ and $n=1$ families are ($-3.563e-2$ and $-7.070e-2$) whereas the fixed boundary target plasma was kink stable to both families.

Of more concern is the fact that when we overlay the port locations on the outboard side of the coil winding surface (**cws**) we find major interferences (overlaps) between the coils and ports:

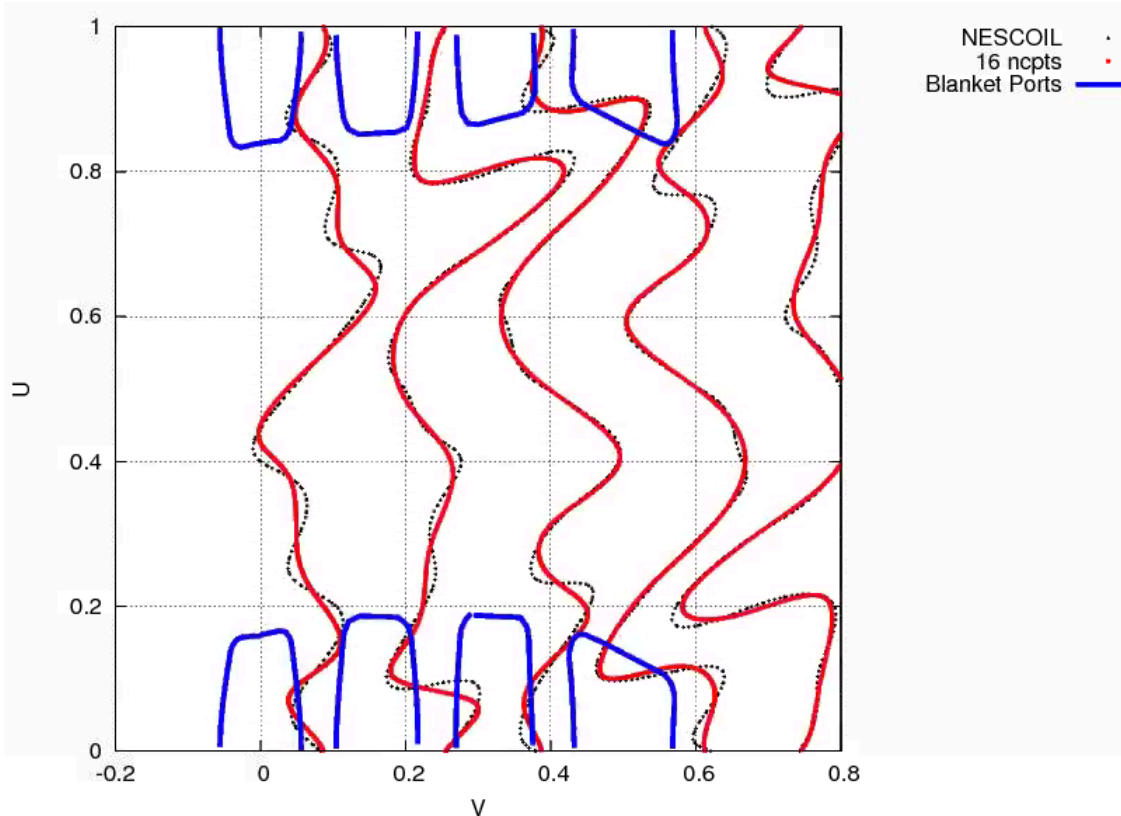


Fig 3

The port geometry shown in Fig 3 were obtained from Tom Brown. Each of the four unique ports (A'-A, A-B, B-C and C-C') are given as a set of x,y,z coordinates which are then mapped to a polygon in u,v coordinates on the winding surface. The true interferences are greater than the above figure suggests. The red and black curves represent coil centerlines and the finite build of the coils has not been taken into account. Our STELLCOPT calculations can take the finite coil build into account since we know the MC coil currents and we can assume a coil current density to provide adjusted polygon vertices in the code which maps from x,y,z to u,v.

In a NESCOIL approach using a non-local Fourier representation for the current potential from which coils are cut it is hard to eliminate local port interferences. But in STELLCOPT, the B-spline representation of coils allows local modification of coil shape. A penalty function for port interference is incorporated, allowing one to ensure that coils will avoid any overlap with a port. Taking account of the finite coil build is trivial.

STELLCOPT avoids the NESCOIL limitation of designing a coilset that reconstructs a particular equilibrium to some a priori unknown degree of precision. STELLCOPT simultaneously designs a plasma and coilset where the physics is desirable and the coils satisfy any imposed engineering constraints. Desirable physics properties are targeted, not the B_n corresponding to a target equilibrium. The detailed plasma shape pops out as part of the solution, simultaneous with coils that support that equilibrium.

Input file for STELLCOPT runs

The input file is identical in form to the input file for free-boundary STELLOPT runs with the exception of additional lines in the **&OPTIMUM** namelist that specify the cws, the spline representation of coils (recall these are produced as output in file **forstellopt** from the **splinefit2** utility), lines that specify the applied engineering constraints, and a separate namelist, **&MGRID_NLI**, that provides parameters for the mgrid calculation.

Let's first look at the COIL_SPLINE info output from the **splinefit2** utility. Make sure you are familiar with the information presented in my B-spline Primer write-up "**B-SPLINE Basics - What you need to know for COIL_SPLINE implementation in STELLCOPT**". Appendix A shows the contents of output file **forstellopt** for the case of 15 control points. The COIL_SPLINESX array for each coil contains the knot values. In this example corresponding to a 15 control point fit to the NESCOIL coils there are $15+4 = 19$ knot values. Notice that the first four knots are set to 0.0, and the last four knots are set to 1.0. This is because we are clamping

both end points to ensure continuity of the uv-coils across $u = 0/1$. The interior knots are uniformly spaced between the end point values. The 15 control points coordinates appear in the COIL_SPLINEFX and COIL_SPLINEFY arrays. COIL_SPLINEFX entries hold the u values for the control points and COIL_SPLINEFY entries hold the corresponding v values. The COIL_SPLINEFX_MIN/MAX and COIL_SPLINEFY_MIN/MAX arrays hold the min/max allowed values for each of the control points. We should set COIL_SPLINEFX_MIN/MAX to be 0/1, the allowed range of u . We can omit mention of COIL_SPLINEFY_MIN/MAX since their values are set by default to -bigno/bigno.

(Note: Presently we must input separate but identical arrays COIL_SPLINESX and COIL_SPLINESY to define the knots. We are in the process of making a code change that will require input of a single array, COIL_SPLINES that will specify the knot locations, and replacement of COIL_SPLINEFX/Y with COIL_SPLINEFU/V. For the moment we will live with the existing notation!)

Figure 4 below shows a plot of the control point locations for just one of the coils (the Type C coil) to show the relationship of the control points and associated polyline with the coil curve:

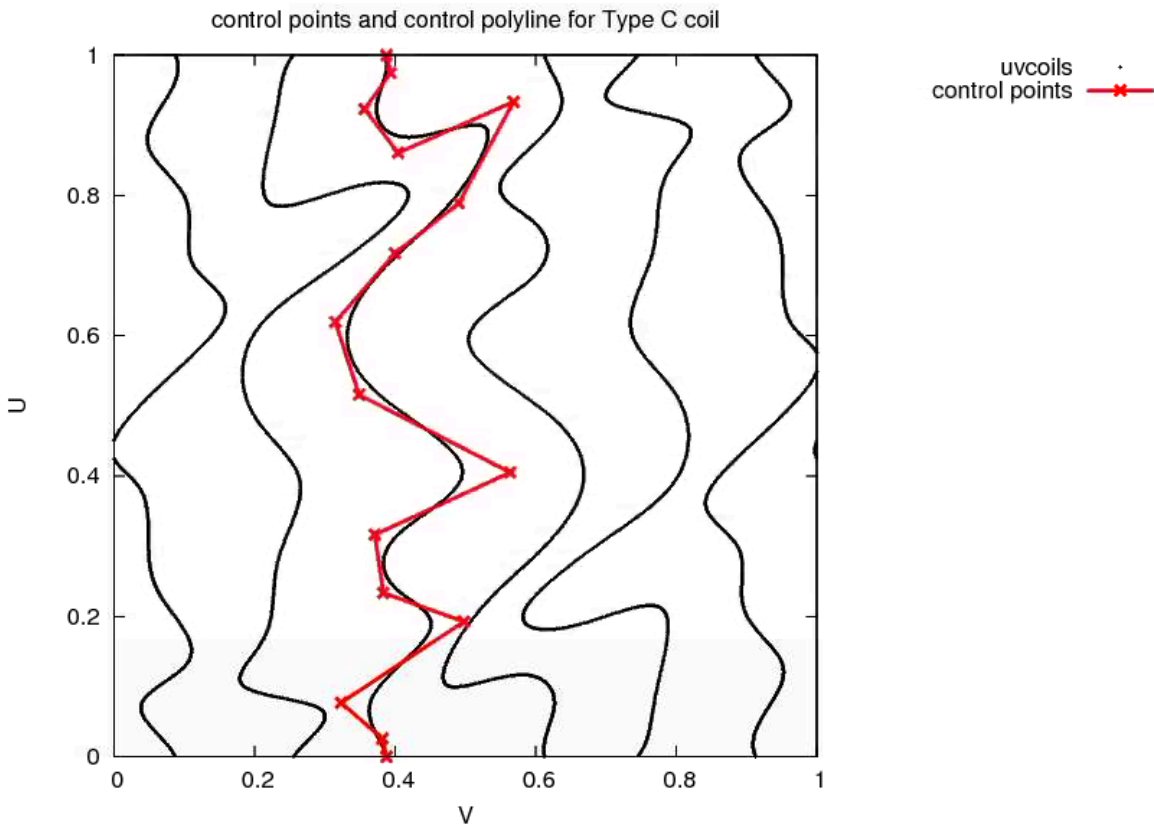


Fig 4

It is pretty clear how one can manipulate the coil curve by hand in order to modify its shape. For example, increasing the v value of the third control point (third from bottom) will straighten the local toroidal excursion of the coil.

To experiment with this by hand, we

1. modify the location of the 3rd control point in the fd.spline to form fd.spline.new
2. generate coils corresponding to the new fd.spline file using syntax

```
/u/jbreslau/bin/spline2coils fd.spline.new cws > coils.new
```

3. generate the coils in uv-space using

```
/u/jbreslau/bin/coils2trunc coils.new cws 3 -o fd.trunc.new
```

4. plot columns 6:5 from the fd.trunc.new file using gnuplot

Here are the old and new control points for the coil#3 (the Type C coil). Only one control point has been changed (see red box):

OLD		NEW	
0.000000e+00	3.874861e-01	0.000000e+00	3.874861e-01
2.564103e-02	3.819303e-01	2.564103e-02	3.819303e-01
7.692308e-02	3.231172e-01	7.692308e-02	3.819303e-01
1.924851e-01	4.976848e-01	1.924851e-01	4.976848e-01
2.335210e-01	3.831122e-01	2.335210e-01	3.831122e-01
3.164558e-01	3.707426e-01	3.164558e-01	3.707426e-01
4.051052e-01	5.635451e-01	4.051052e-01	5.635451e-01
5.159949e-01	3.485475e-01	5.159949e-01	3.485475e-01
6.196003e-01	3.146485e-01	6.196003e-01	3.146485e-01
7.175007e-01	3.994811e-01	7.175007e-01	3.994811e-01
7.887017e-01	4.899417e-01	7.887017e-01	4.899417e-01
9.334742e-01	5.682675e-01	9.334742e-01	5.682675e-01
8.606337e-01	4.041521e-01	8.606337e-01	4.041521e-01
9.230769e-01	3.564515e-01	9.230769e-01	3.564515e-01
9.743590e-01	3.930418e-01	9.743590e-01	3.930418e-01
1.000000e+00	3.874861e-01	1.000000e+00	3.874861e-01

In the uvcoil plot, Fig 5 below, we note the local modification of the coil (black coil curve ---> blue coil curve) as a result of changing the u value of the 3rd control point.

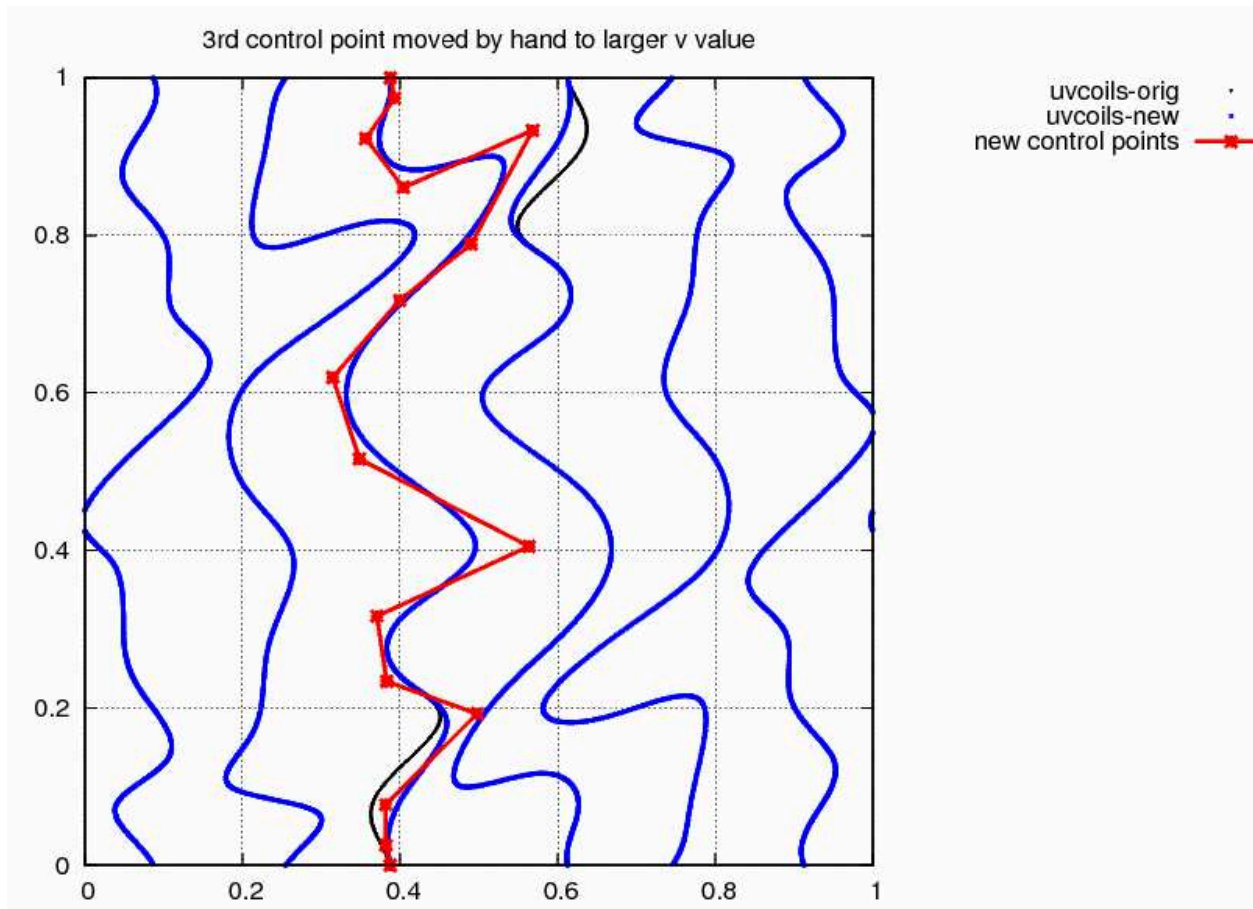


Fig 5

Of course, we use STELLCOPT to effect such modifications automatically rather than doing it by hand. But sometimes it is necessary to perform a local modification by hand.

STELLCOPT input file lines which control coil splines and various engineering constraints:

Appendix B gives a reasonably complete set of input file lines that deal with the coil splines

`WINDSURFNAME = 'cws'` - make sure the file containing the coil windingsurface Fourier components (NESCOIL convention) is in area that STELLCOPT is being executed. In this case the filename is "cws"

`COIL_TYPE(001) = 'M'` - specifies which boundary conditions are being used for the coils #1. Use 'M' unless we are forcing pure straightening over a poloidal range when we will use 'A'. This will be discussed at some point in the future.

`LCOIL_SPLINE(001,:) = 15*T` - logical array specifying which of the control point locations is allowed to vary during the STELLCOPT iterations. Any control point can be fixed by choosing its corresponding array element to be F. For example `LCOIL_SPLINE(001,:) = F 13*T F` would fix the toroidal position of coil #1 on the outboard midplane where $u = 0/1$.

`COIL_SPLINESX(001,:) =` knot positions for coil curve 1. Equally spaced, and with clamping

`COIL_SPLINESY(001,:) =` knot positions for coil curve 1. Repeat of `COIL_SPLINESX`

`COIL_SPLINEFX(001,:) =` u coordinates of control points for coil curve 1

`COIL_SPLINEFY(001,:) =` v coordinates of control points for coil curve 1

`COIL_SPLINEFX_MIN(001,:) =` min allowed u coord of control pts. Set all equal to 0.0

`COIL_SPLINEFX_MAX(001,:) =` max allowed u coord of control pts. Set all elements = 1.0

`COIL_SPLINEFY_MIN(001,:) =` min allowed v coord of control pts. No need to set.

`COIL_SPLINEFY_MAX(001,:) =` max allowed v coord of control pts. No need to set.

Etc for coil curves 2 and 3

`TARGET_COILCRV(001) =` Target value for maximum coil curvature [m^{-1}] of coil curve 1

`SIGMA_COILCRV(001) =` Inverse weight for enforcing curvature constraint on coil curve 1

Etc for coil curves 2 and 3

`TARGET_COILSEP =` Target value for minimum coil-to-coil separation between any 2 coils.

`SIGMA_COILSEP =` Inverse weight for enforcing minimum coil-to-coil separation.

`KOPOLYU(:, 001) =` u coords defining vertices of polygonal stayout zone defining 1st port

`KOPOLYV(:, 001) =` v coords defining vertices of polygonal stayout zone.

`TARGET_COILPOLY(001) =` set this to be 0 (count of how many points on coil curve are allowed inside a keepout zone)

`SIGMA_COILPOLY(001) =` set this to be 0.1 to give a strong penalty

Etc for all other ports or coil keepout zones

Two Illustrative Runs

Recall that Fig 3 showed that the NESCOIL cut coils had passed through port regions essential for stellarator maintenance. No matter how good the physics of plasmas supported by such coils might be, the resulting configuration is physically irrelevant!

To obtain a starting configuration that is related to the NESCOIL geometry, but which satisfies maintenance through the ports, let us manually manipulate a few of the control points near $u = 0/1$ so that the coil curves pass between the ports and show no overlap with these keep-out regions. All we are doing is to come up with a starting coil configuration - one for which there is a converged free-boundary VMEC solution - allowing STELLCOPT to begin iterating. If we take too much liberty with the control point manipulation we can expect to find problems with VMEC convergence. However, we find that VMEC convergence is quite robust with respect to local coil shape manipulation in the outboard midplane region of the CWS so this is not a big problem. While VMEC convergence is robust, we make no such claims for the physics of the corresponding free-boundary plasma. Whatever physics we destroy with such manipulation we hope to recover using STELLCOPT.

Figure 6 shows the coils before (black) and after (red) the hand-manipulation of control points.

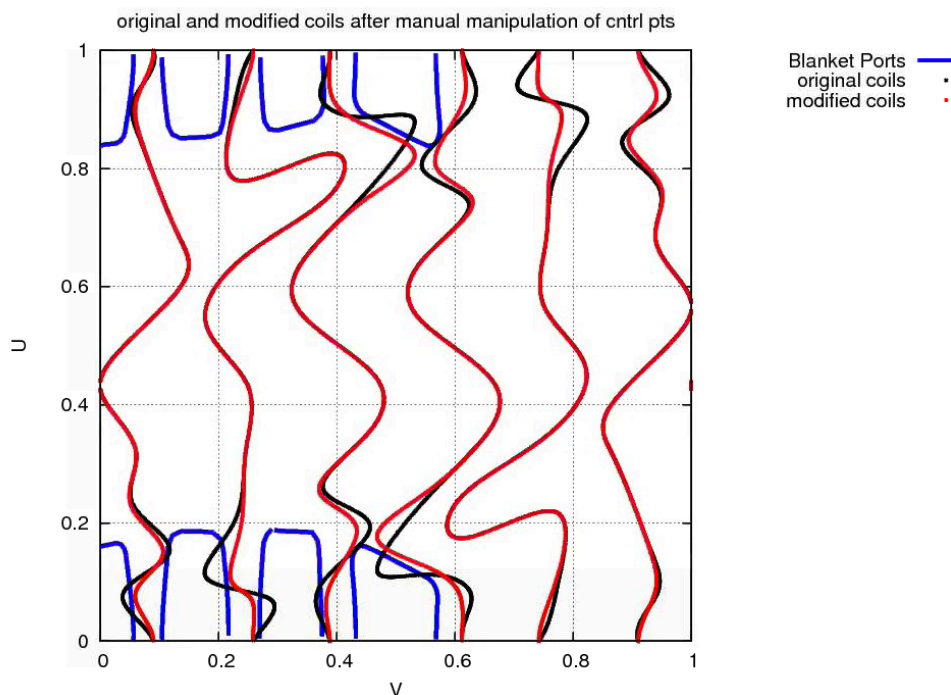


Fig 6

The original and modified control points are listed below. I used our knowledge of local control of B-spline curves to effect the change. Of course there are an infinite number of solutions that effect the change we are after. I stress that we are simply coming up with an initial spline curve that satisfies the port keep-out constraints.

Original control points

#	coil A		coil B		coil C	
	u	v	u	v	u	v
0.000000e+00	8.964011e-02	0.000000e+00	2.581017e-01	0.000000e+00	3.880744e-01	
2.777778e-02	8.001243e-02	2.777778e-02	2.765970e-01	2.777778e-02	3.837616e-01	
8.333333e-02	-1.896723e-02	8.333333e-02	3.403251e-01	8.333333e-02	3.196130e-01	
1.491114e-01	1.753708e-01	8.993240e-02	1.218641e-01	2.080381e-01	5.251133e-01	
2.278207e-01	1.725189e-02	1.984065e-01	2.488615e-01	2.449275e-01	3.182762e-01	
3.219772e-01	8.829397e-02	3.182152e-01	2.360821e-01	3.471373e-01	4.663598e-01	
4.095844e-01	-9.719242e-03	4.389317e-01	2.793190e-01	4.497146e-01	5.070915e-01	
4.723388e-01	-9.301214e-04	5.358796e-01	1.359628e-01	5.669277e-01	2.764104e-01	
6.157407e-01	1.719553e-01	6.678362e-01	2.427018e-01	6.855845e-01	3.737079e-01	
6.768049e-01	1.344120e-01	7.521495e-01	4.199546e-01	7.520620e-01	4.476147e-01	
7.665267e-01	9.761359e-02	8.689768e-01	4.240287e-01	9.140770e-01	5.769538e-01	
8.577067e-01	6.477984e-02	7.242175e-01	1.852975e-01	8.731549e-01	4.357309e-01	
9.166667e-01	3.879885e-02	9.166667e-01	2.293534e-01	9.166667e-01	3.454902e-01	
9.722222e-01	9.926779e-02	9.722222e-01	2.396064e-01	9.722222e-01	3.923873e-01	
1.000000e+00	8.964011e-02	1.000000e+00	2.581017e-01	1.000000e+00	3.880744e-01	

Modified Control points to produce a coil curve that does not interfere with the ports

#	coil A		coil B		coil C	
	u	v	u	v	u	v
0.000000e+00	8.964011e-02	0.000000e+00	2.581017e-01	0.000000e+00	3.880744e-01	
2.777778e-02	8.964011e-02	2.777778e-02	2.581017e-01	2.777778e-02	3.880744e-01	
8.333333e-02	0.02	8.333333e-02	0.27	8.333333e-02	0.36	
1.491114e-01	0.15	8.993240e-02	0.20	2.080381e-01	0.48	
2.278207e-01	1.725189e-02	1.984065e-01	2.488615e-01	2.449275e-01	3.182762e-01	
3.219772e-01	8.829397e-02	3.182152e-01	2.360821e-01	3.471373e-01	4.663598e-01	
4.095844e-01	-9.719242e-03	4.389317e-01	2.793190e-01	4.497146e-01	5.070915e-01	
4.723388e-01	-9.301214e-04	5.358796e-01	1.359628e-01	5.669277e-01	2.764104e-01	
6.157407e-01	1.719553e-01	6.678362e-01	2.427018e-01	6.855845e-01	3.737079e-01	
6.768049e-01	1.344120e-01	7.521495e-01	4.199546e-01	7.520620e-01	4.476147e-01	
7.665267e-01	9.761359e-02	8.689768e-01	4.240287e-01	0.83	5.769538e-01	
8.577067e-01	6.477984e-02	7.242175e-01	1.852975e-01	8.731549e-01	4.357309e-01	
9.166667e-01	0.05	9.166667e-01	2.293534e-01	9.166667e-01	0.38	
9.722222e-01	8.964011e-02	9.722222e-01	2.581017e-01	9.722222e-01	3.880744e-01	
1.000000e+00	8.964011e-02	1.000000e+00	2.581017e-01	1.000000e+00	3.880744e-01	

The first of the illustrative STELLCOPT runs (RUN1) begins from a coilset and equilibrium corresponding to the modified control points above. We run for 1542 iterations (FUNC_MAX = 1500), equivalent to 17 "major iterations" of LMDIF for the given degrees of freedom of the problem. The input file for this case is named input.modified and can be examined in

/p/stellopt/ANALYSIS/Pomphrey/LDRD20/STELLOPT_TUTORIAL/CHANGE_INITIAL_CNTRL_PTS/RUN1

Active physics targets are: VOLUME, BETA and ASPECT applied with modest weights (SIGMA VALUES = 1.0), HELICITY, NEO(025), KINK(001) and KINK(002) with medium weights (SIGMA = 0.1) and IOTA(049) with a strong weight (SIGMA = 0.01). The initial equilibrium value of edge iota is an extremely low value of 0.32 and the point of this run is to see how the coil shapes change when we demand a TARGET_IOTA value on edge of 0.55

Active engineering targets are COILCRV and COILSEP with modest weight (SIGMA = 1.0). The keepout polygon constraints ARE NOT APPLIED.

Figure 7a below show the initial (black) and final coil shapes (red) on the u-v plane. The final iota value for the equilibrium corresponding to iteration 01542 is found to equal the target value of 0.55. The $\epsilon_{eff}^{3/2}$ value calculated by NEO is decreased by approximately a factor of two during the iterations and the n=0 kink family has a modestly improved unstable eigenvalue. However, we note that the physics targets have improved at the expense of violating the desire to have a coilset that does not interfere with the ports.

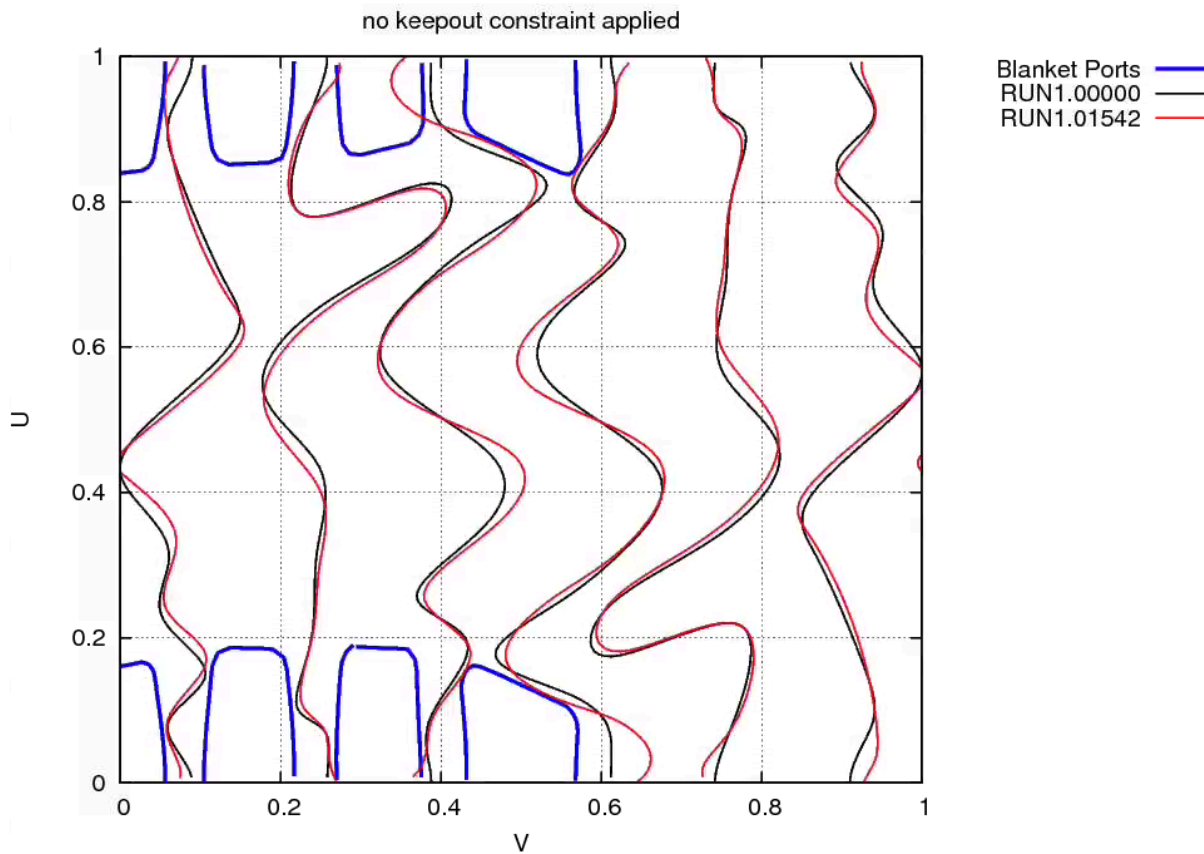


Fig 7a

So in the second illustrative run (RUN2) we keep the same physics and engineering targets in the input file - see input file input.modified in /p/stellopt/ANALYSIS/Pomphrey/LDRD20/STELLOPT_TUTORIAL/CHANGE_INITIAL_CNTRL_PTS/RUN2 - but apply the keep-out polygon constraint. Whereas in RUN1 we set all the SIGMA_COILPOLY inverse weights to be 1.e8, we set them equal to 0.1 in RUN2. Apart from this change, the input files for RUN2 and RUN1 are identical.

Figure 7b below shows the initial and final coil curves in the CWS u-v plane

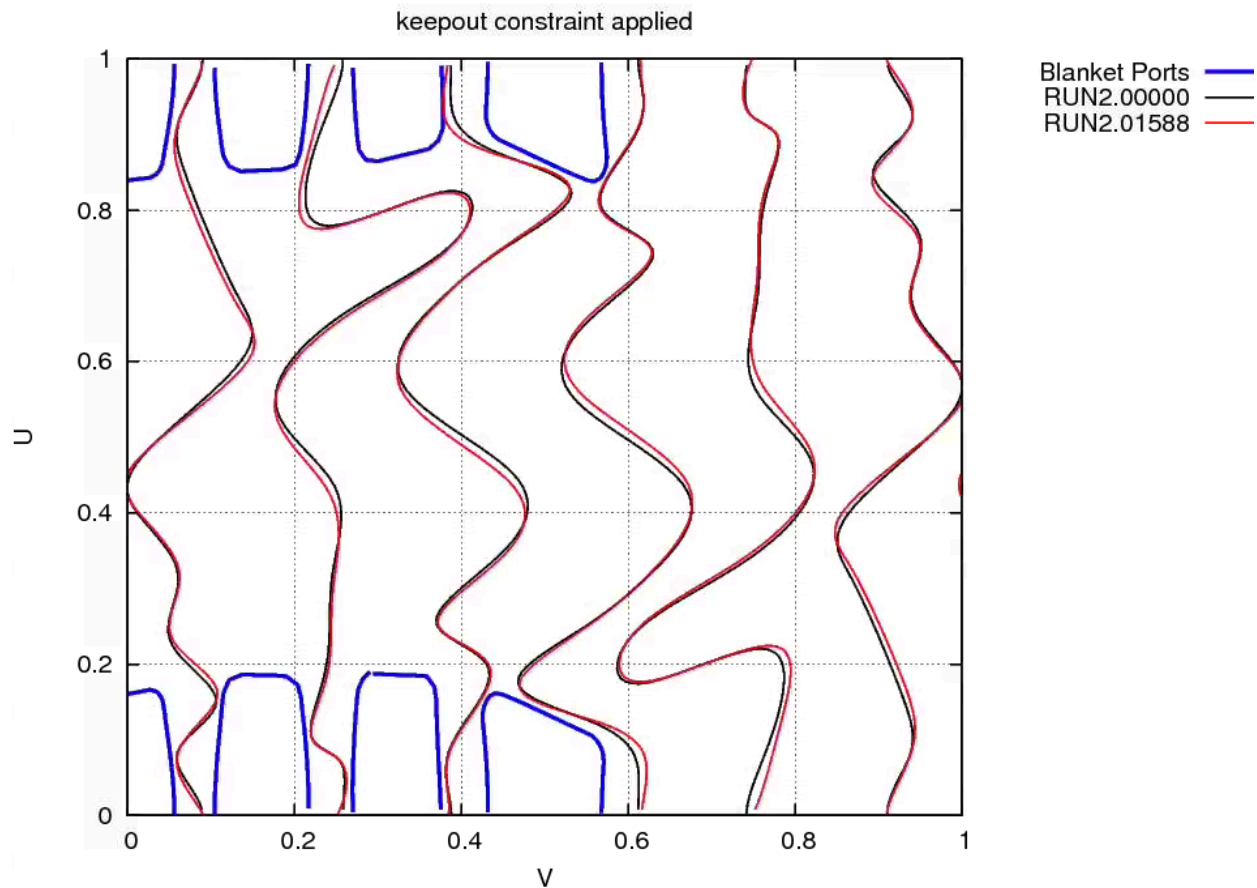


Fig 7b

Note that the coil centroids remain outside the keep-out port zones (there is a point of essential tangency of the Type C coil (red curve, third from left) with B-C port (blue, third from left)).

Figs 7c, d, e show the evolution of the three of the physics targets for RUN1 and RUN2.

Fig 7c shows the evolution of edge iota as a function of "minor iteration" number. Fig 7d shows the evolution of $\epsilon_{eff}^{3/2}$ at surface 25 out of 49. Fig 7e shows the evolution of the n=0 and n=1 kink family eigenvalues.

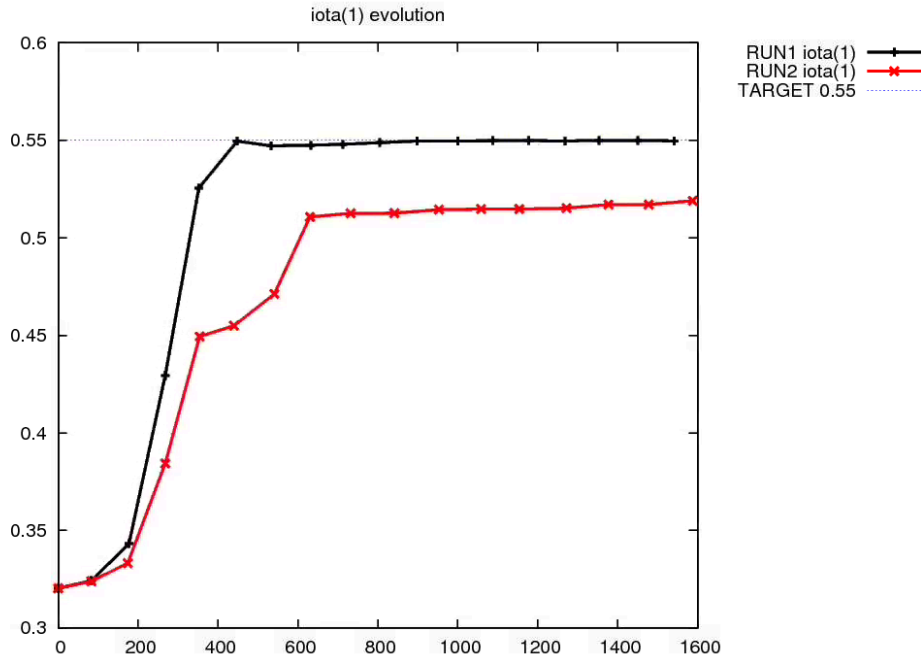


Fig 7c

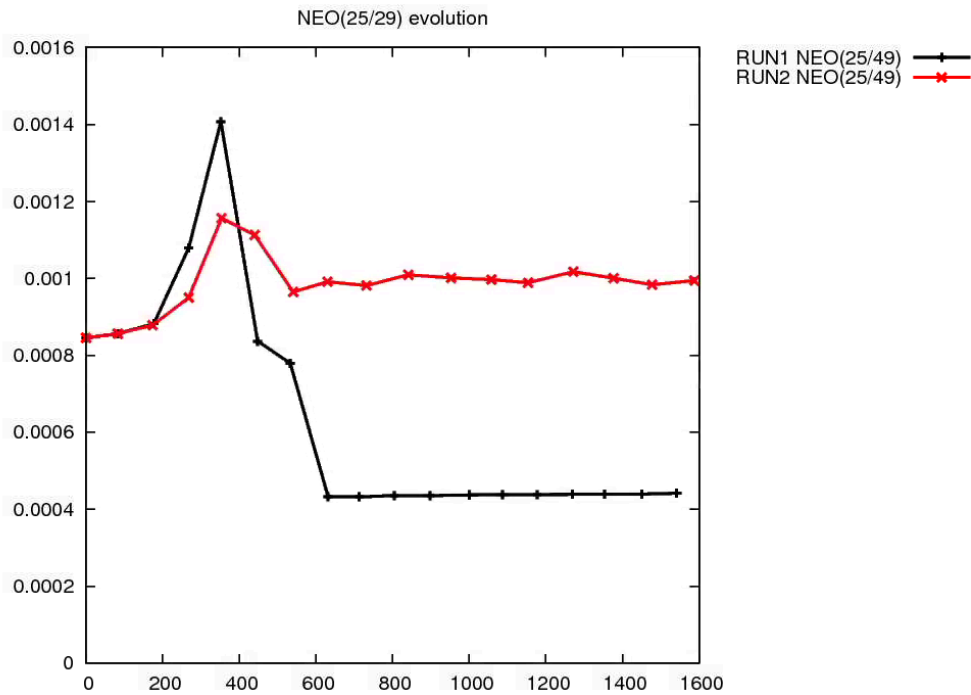


Fig 7d

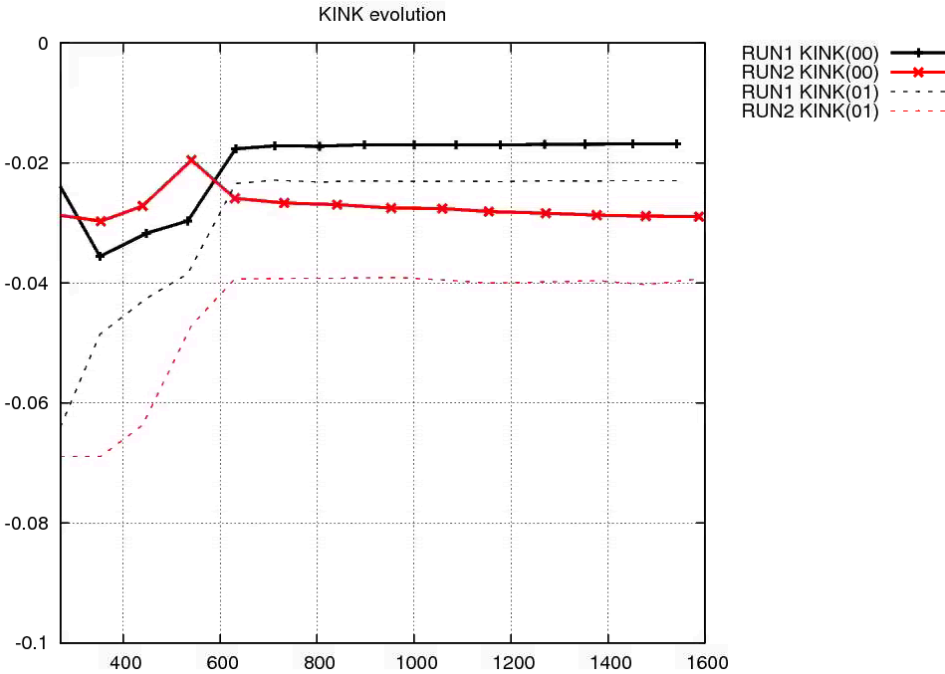


Fig 7e

RUN2 has not met the iota target value of 0.55. It is getting there, but only slowly. The $\epsilon_{eff}^{3/2}$ value for RUN2 is degraded slightly with respect to the initial value (we should not be comparing with the final state of RUN1 because this run is unacceptable from an engineering standpoint!). The n=0 kink modes show little improvement but the n=1 family is definitely improved. However, I would regard any eigenvalue greater than 1.e-2 as a rather unstable eigenvalue. So these KINK values are rather unacceptable.

So how would we proceed from here? We would develop strategies for improving all the physics by adjusting the relative weights of the various targets. That is for the user to experiment with and develop their own strategies. I prefer to focus mainly on one important target and find a weight distribution that improves it with little or no expense to other targets. Cycle through the desired targets and when you have achieved a good target, and move to a new target (or targets) try to work with the minimally effective weight (largest SIGMA) for the acceptable target.

Using essentially the same physics targets as above, I have achieved a configuration with $\epsilon_{eff}^{3/2} = 4.5e-4$ (approx the same as the engineering unacceptable RUN1 in Fig 7d) with KINK eigenvalues of $-5.5e-4$ and $-6.9e-3$ (much improved from Fig 7e above) and with good vacuum

surface (not discussed above). Fig 8, below, shows this configuration, and it is clear that this coil configuration is rather different from the RUN2.

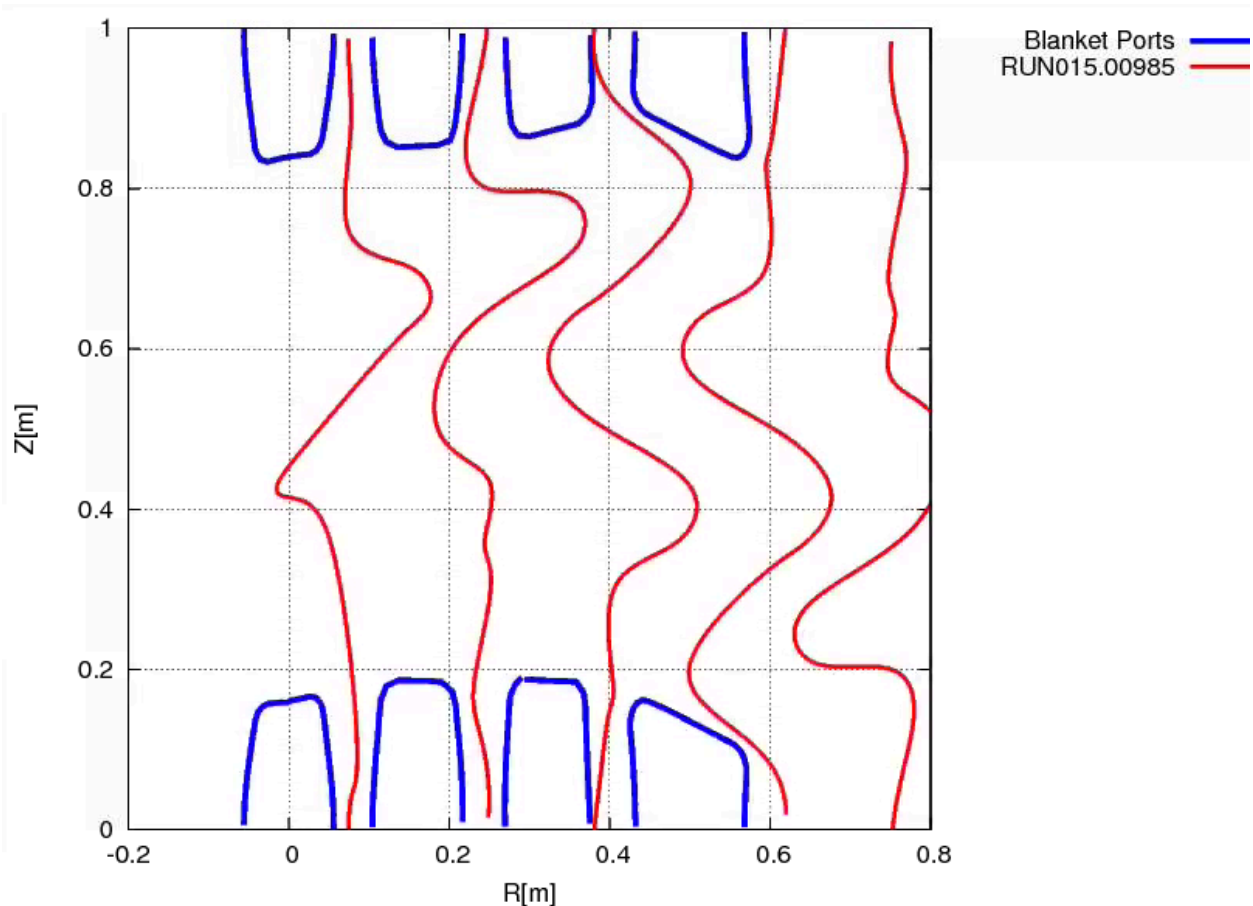


Fig 8

As I have said many times, it is important not to think of STELLOPT or STELLCOPT as a black box that you turn on and hope to get a good solution. The user must steer it toward a desirable solution.

The executable for STELLCOPT (xstelloptv2) can be copied from </p/m3d/jbreslau/Stellarator/Github/STELLOPT/STELLOPTV2/Release/>

A typical batchjob I use for submitting runs is

```
#!/bin/csh
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH -J modified
#SBATCH --mem=8GB
#SBATCH --time=12:00:00
#SBATCH -e logfile.modified
#SBATCH --mail-type=ALL
#SBATCH --mail-user=pomphrey@pppl.gov
```

```
source ~/jbreslau/gstelmod
mpirun -np 16 ./xstelloptv2 input.modified
```

Typically I run with `FUNC_MAX = 500` or `1000`, look at results, and re-adjust weights for the next run.

Appendix A: Example of `forstellopt` file produced by `splinefit2` using 15 control points (hence 19 knots). See main text for explanation.

```
!----- COIL 0001
COIL_SPLINESX(1,:) = 0.0000000e+00 0.0000000e+00 0.0000000e+00
    0.0000000e+00 8.3333333e-02 1.6666667e-01
    2.5000000e-01 3.3333333e-01 4.1666667e-01
    5.0000000e-01 5.8333333e-01 6.6666667e-01
    7.5000000e-01 8.3333333e-01 9.1666667e-01
    1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00
COIL_SPLINEFX_MIN(1,:) = 15*0.0000000e+00
COIL_SPLINEFX(1,:) = 0.0000000e+00 2.777778e-02 8.333333e-02
    1.491114e-01 2.278207e-01 3.219772e-01
    4.095844e-01 4.723388e-01 6.157407e-01
    6.768049e-01 7.665267e-01 8.577067e-01
    9.166667e-01 9.722222e-01 1.000000e+00
COIL_SPLINEFX_MAX(1,:) = 15*1.0000000e+00
COIL_SPLINEFY_MIN(1,:) = 15*-6.569818e-02
COIL_SPLINEFY(1,:) = 8.964011e-02 8.001243e-02 -1.896723e-02
    1.753708e-01 1.725189e-02 8.829397e-02
    -9.719242e-03 -9.301214e-04 1.719553e-01
    1.344120e-01 9.761359e-02 6.477984e-02
    3.879885e-02 9.926779e-02 8.964011e-02
COIL_SPLINEFY_MAX(1,:) = 15*2.258088e-01
!----- COIL 0002
COIL_SPLINESX(2,:) = 0.0000000e+00 0.0000000e+00 0.0000000e+00
    0.0000000e+00 8.3333333e-02 1.6666667e-01
    2.5000000e-01 3.3333333e-01 4.1666667e-01
    5.0000000e-01 5.8333333e-01 6.6666667e-01
    7.5000000e-01 8.3333333e-01 9.1666667e-01
    1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00
COIL_SPLINEFX_MIN(2,:) = 15*0.0000000e+00
```

```

COIL_SPLINEFX(2,:) = 0.000000e+00 2.777778e-02 8.333333e-02
      8.993240e-02 1.984065e-01 3.182152e-01
      4.389317e-01 5.358796e-01 6.678362e-01
      7.521495e-01 8.689768e-01 7.242175e-01
      9.166667e-01 9.722222e-01 1.000000e+00
COIL_SPLINEFX_MAX(2,:) = 15*1.000000e+00
COIL_SPLINEFY_MIN(2,:) = 15*5.292427e-02
COIL_SPLINEFY(2,:) = 2.581017e-01 2.765970e-01 3.403251e-01
      1.218641e-01 2.488615e-01 2.360821e-01
      2.793190e-01 1.359628e-01 2.427018e-01
      4.199546e-01 4.240287e-01 1.852975e-01
      2.293534e-01 2.396064e-01 2.581017e-01
COIL_SPLINEFY_MAX(2,:) = 15*5.061711e-01
!----- COIL 0003
COIL_SPLINESX(3,:) = 0.000000e+00 0.000000e+00 0.000000e+00
      0.000000e+00 8.333333e-02 1.666667e-01
      2.500000e-01 3.333333e-01 4.166667e-01
      5.000000e-01 5.833333e-01 6.666667e-01
      7.500000e-01 8.333333e-01 9.166667e-01
      1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
COIL_SPLINEFX_MIN(3,:) = 15*0.000000e+00
COIL_SPLINEFX(3,:) = 0.000000e+00 2.777778e-02 8.333333e-02
      2.080381e-01 2.449275e-01 3.471373e-01
      4.497146e-01 5.669277e-01 6.855845e-01
      7.520620e-01 9.140770e-01 8.731549e-01
      9.166667e-01 9.722222e-01 1.000000e+00
COIL_SPLINEFX_MAX(3,:) = 15*1.000000e+00
COIL_SPLINEFY_MIN(3,:) = 15*2.097936e-01
COIL_SPLINEFY(3,:) = 3.880744e-01 3.837616e-01 3.196130e-01
      5.251133e-01 3.182762e-01 4.663598e-01
      5.070915e-01 2.764104e-01 3.737079e-01
      4.476147e-01 5.769538e-01 4.357309e-01
      3.454902e-01 3.923873e-01 3.880744e-01
COIL_SPLINEFY_MAX(3,:) = 15*6.606088e-01

```

Appendix B: - lines which control various engineering constraints:

The following lines will typically appear in the &OPTIMUM namelist.

Note that this example pertains to a case with 15 control points and 19 knots. It is not the same as in Appendix A.

```

!-----
!   COIL TARGETS and ENGINEERING CONSTRAINTS
!-----
!
WINDSURFNAME = 'cws'
!-----
!   COIL SPLINES
!-----
!   Coil Number 001
COIL_TYPE(001) = 'M'
LCOIL_SPLINE(001,:) = 15*T
COIL_SPLINESX(001,:) = 0.000000000000E+000 0.000000000000E+000 0.000000000000E+000
      0.000000000000E+000 8.333333000000E-002 1.666667000000E-001
      2.500000000000E-001 3.333333000000E-001 4.166667000000E-001
      5.000000000000E-001 5.833333000000E-001 6.666667000000E-001
      7.500000000000E-001 8.333333000000E-001 9.166667000000E-001
      1.000000000000E+000 1.000000000000E+000 1.000000000000E+000
      1.000000000000E+000

```

```

COIL_SPLINESY( 001,;) = 0.00000000000E+000 0.00000000000E+000 0.00000000000E+000
0.00000000000E+000 8.33333300000E-002 1.66666700000E-001
2.50000000000E-001 3.33333300000E-001 4.16666700000E-001
5.00000000000E-001 5.83333300000E-001 6.66666700000E-001
7.50000000000E-001 8.33333300000E-001 9.16666700000E-001
1.00000000000E+000 1.00000000000E+000 1.00000000000E+000
1.00000000000E+000

COIL_SPLINEFX( 001,;) = 0.00000000000E+000 4.757453253133E-002 4.038119035936E-002
1.435975065326E-001 3.998345455087E-001 4.151852688223E-001
4.180235410912E-001 4.939981097075E-001 6.289414344272E-001
6.515131149236E-001 7.109670028060E-001 7.168934423281E-001
8.812649194807E-001 9.524254845955E-001 1.00000000000E+000

COIL_SPLINEFY( 001,;) = 7.433733813820E-002 7.524540826115E-002 8.560815460583E-002
8.686326168333E-002 5.441078584210E-002 5.544699553805E-003
-3.137543092663E-002 3.328566631329E-002 1.468634695919E-001
1.856207530735E-001 1.666297576555E-001 5.108791298852E-002
8.587437009906E-002 7.342926834215E-002 7.433733813820E-002

COIL_SPLINEFX_MIN( 001,;) = 0.00000000000E+000 0.00000000000E+000 0.00000000000E+000
0.00000000000E+000 0.00000000000E+000 0.00000000000E+000
0.00000000000E+000 0.00000000000E+000 0.00000000000E+000
0.00000000000E+000 0.00000000000E+000 0.00000000000E+000
0.00000000000E+000 0.00000000000E+000 0.00000000000E+000

COIL_SPLINEFX_MAX( 001,;) = 1.00000000000E+000 1.00000000000E+000 1.00000000000E+000
1.00000000000E+000 1.00000000000E+000 1.00000000000E+000
1.00000000000E+000 1.00000000000E+000 1.00000000000E+000
1.00000000000E+000 1.00000000000E+000 1.00000000000E+000
1.00000000000E+000 1.00000000000E+000 1.00000000000E+000

COIL_SPLINEFY_MIN( 001,;) = -1.00000000000E-001 -1.00000000000E-001 -1.00000000000E-001
-1.00000000000E-001 -1.00000000000E-001 -1.00000000000E-001
-1.00000000000E-001 -1.00000000000E-001 -1.00000000000E-001
-1.00000000000E-001 -1.00000000000E-001 -1.00000000000E-001
-1.00000000000E-001 -1.00000000000E-001 -1.00000000000E-001

COIL_SPLINEFY_MAX( 001,;) = 2.75000000000E-001 2.75000000000E-001 2.75000000000E-001
2.75000000000E-001 2.75000000000E-001 2.75000000000E-001
2.75000000000E-001 2.75000000000E-001 2.75000000000E-001
2.75000000000E-001 2.75000000000E-001 2.75000000000E-001
2.75000000000E-001 2.75000000000E-001 2.75000000000E-001

!-----
! Coil Number 002
COIL_TYPE( 002) = 'M'
LCOIL_SPLINE( 002,;) = 15*T
COIL_SPLINESX( 002,;) = 0.00000000000E+000 0.00000000000E+000 0.00000000000E+000
0.00000000000E+000 8.33333300000E-002 1.66666700000E-001
2.50000000000E-001 3.33333300000E-001 4.16666700000E-001
5.00000000000E-001 5.83333300000E-001 6.66666700000E-001
7.50000000000E-001 8.33333300000E-001 9.16666700000E-001
1.00000000000E+000 1.00000000000E+000 1.00000000000E+000
1.00000000000E+000

COIL_SPLINESY( 002,;) = 0.00000000000E+000 0.00000000000E+000 0.00000000000E+000
0.00000000000E+000 8.33333300000E-002 1.66666700000E-001
2.50000000000E-001 3.33333300000E-001 4.16666700000E-001
5.00000000000E-001 5.83333300000E-001 6.66666700000E-001
7.50000000000E-001 8.33333300000E-001 9.16666700000E-001
1.00000000000E+000 1.00000000000E+000 1.00000000000E+000
1.00000000000E+000

COIL_SPLINEFX( 002,;) = 0.00000000000E+000 5.925864452132E-002 1.379163947408E-001
1.730928517467E-001 3.300814818057E-001 3.466103016673E-001
4.503563728857E-001 4.724271156304E-001 6.365229704615E-001
7.185651045871E-001 8.125614056924E-001 7.819408764158E-001
8.545004277583E-001 9.407413768118E-001 1.00000000000E+000

COIL_SPLINEFY( 002,;) = 2.473601853284E-001 2.542154227959E-001 2.335369119151E-001
2.242503148395E-001 2.642246863265E-001 2.316394447502E-001
2.747670455024E-001 1.642432760389E-001 1.923352214589E-001
3.915352627596E-001 3.599716514800E-001 2.350194435031E-001

```

```

2.088308860032E-001  2.405049503288E-001  2.473601853284E-001
COIL_SPLINEFX_MIN( 002,:) = 0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
COIL_SPLINEFX_MAX( 002,:) = 1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
COIL_SPLINEFY_MIN( 002,:) = 7.500000000000E-002  7.500000000000E-002  7.500000000000E-002
7.500000000000E-002  7.500000000000E-002  7.500000000000E-002
7.500000000000E-002  7.500000000000E-002  7.500000000000E-002
7.500000000000E-002  7.500000000000E-002  7.500000000000E-002
7.500000000000E-002  7.500000000000E-002  7.500000000000E-002
COIL_SPLINEFY_MAX( 002,:) = 5.000000000000E-001  5.000000000000E-001  5.000000000000E-001
5.000000000000E-001  5.000000000000E-001  5.000000000000E-001
5.000000000000E-001  5.000000000000E-001  5.000000000000E-001
5.000000000000E-001  5.000000000000E-001  5.000000000000E-001
5.000000000000E-001  5.000000000000E-001  5.000000000000E-001

```

```
!-----
```

```
!   Coil Number 003
```

```
COIL_TYPE( 003) = 'M'
```

```
LCOIL_SPLINE( 003,:) = 15*T
```

```

COIL_SPLINESX( 003,:) = 0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  8.333333000000E-002  1.666667000000E-001
2.500000000000E-001  3.333333000000E-001  4.166667000000E-001
5.000000000000E-001  5.833333000000E-001  6.666667000000E-001
7.500000000000E-001  8.333333000000E-001  9.166667000000E-001
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000
COIL_SPLINESY( 003,:) = 0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  8.333333000000E-002  1.666667000000E-001
2.500000000000E-001  3.333333000000E-001  4.166667000000E-001
5.000000000000E-001  5.833333000000E-001  6.666667000000E-001
7.500000000000E-001  8.333333000000E-001  9.166667000000E-001
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000
COIL_SPLINEFX( 003,:) = 0.000000000000E+000  6.936548834631E-002  1.651376265072E-001
1.577379986833E-001  3.260181659427E-001  3.454735849832E-001
4.343154995066E-001  5.052240585695E-001  5.853031647304E-001
6.514186883376E-001  6.620158597896E-001  8.038705778269E-001
8.686242525875E-001  9.306345366253E-001  1.000000000000E+000
COIL_SPLINEFY( 003,:) = 3.808384132129E-001  3.889434395414E-001  3.984865819817E-001
4.106586941869E-001  3.826075394062E-001  4.936563461001E-001
5.321497206585E-001  3.677618199881E-001  3.031490844167E-001
3.600086687061E-001  3.922507306220E-001  5.376373386596E-001
4.525943294166E-001  3.727326119157E-001  3.808384132129E-001
COIL_SPLINEFX_MIN( 003,:) = 0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
0.000000000000E+000  0.000000000000E+000  0.000000000000E+000
COIL_SPLINEFX_MAX( 003,:) = 1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
1.000000000000E+000  1.000000000000E+000  1.000000000000E+000
COIL_SPLINEFY_MIN( 003,:) = 1.250000000000E-001  1.250000000000E-001  1.250000000000E-001
1.250000000000E-001  1.250000000000E-001  1.250000000000E-001
1.250000000000E-001  1.250000000000E-001  1.250000000000E-001
1.250000000000E-001  1.250000000000E-001  1.250000000000E-001
1.250000000000E-001  1.250000000000E-001  1.250000000000E-001

```

```

COIL_SPLINEFY_MAX( 003,:) = 7.250000000000E-001  7.250000000000E-001  7.250000000000E-001
      7.250000000000E-001  7.250000000000E-001  7.250000000000E-001
      7.250000000000E-001  7.250000000000E-001  7.250000000000E-001
      7.250000000000E-001  7.250000000000E-001  7.250000000000E-001

```

```

!-----
! COIL CURVATURE CONSTRAINTS
!-----

```

```

TARGET_COILCRV( 001) = 1.000000000000E+000
SIGMA_COILCRV( 001) = 1.000000000000E+000
TARGET_COILCRV( 002) = 1.000000000000E+000
SIGMA_COILCRV( 002) = 1.000000000000E+000
TARGET_COILCRV( 003) = 1.000000000000E+000
SIGMA_COILCRV( 003) = 1.000000000000E+000
!

```

```

!-----
! MINIMUM COIL-to-COIL SEPARATION CONSTRAINT
!-----

```

```

TARGET_COILSEP = 6.500000000000E-001
SIGMA_COILSEP = 1.000000000000E+000
!

```

```

!-----
! COIL EXCLUSION ZONE (eg PORTS)
!-----

```

```

! Keepout Polygon 001

```

```

KOPOLYU(:, 001) = 1.000000000000E+000  9.533000000000E-001  9.066000000000E-001
      8.667000000000E-001  8.505000000000E-001  8.427000000000E-001
      8.362000000000E-001  8.337000000000E-001  8.358000000000E-001
      8.421000000000E-001  8.432000000000E-001  8.475000000000E-001
      8.539000000000E-001  8.611000000000E-001  8.758000000000E-001
      8.977000000000E-001  9.194000000000E-001  9.413000000000E-001
      9.635000000000E-001  1.000000000000E+000
KOPOLYV(:, 001) = -5.600000000000E-002  -5.410000000000E-002  -4.940000000000E-002
      -4.450000000000E-002  -4.230000000000E-002  -3.980000000000E-002
      -3.490000000000E-002  -2.670000000000E-002  -1.780000000000E-002
      1.780000000000E-002  2.680000000000E-002  3.400000000000E-002
      3.860000000000E-002  4.120000000000E-002  4.370000000000E-002
      4.710000000000E-002  5.020000000000E-002  5.290000000000E-002
      5.480000000000E-002  5.600000000000E-002

```

```

! Keepout Polygon 002

```

```

KOPOLYU(:, 002) = 0.000000000000E+000  2.910000000000E-002  5.870000000000E-002
      8.780000000000E-002  1.169000000000E-001  1.316000000000E-001
      1.461000000000E-001  1.525000000000E-001  1.568000000000E-001
      1.590000000000E-001  1.642000000000E-001  1.663000000000E-001
      1.638000000000E-001  1.573000000000E-001  1.495000000000E-001
      1.333000000000E-001  1.013000000000E-001  6.990000000000E-002
      3.900000000000E-002  0.000000000000E+000
KOPOLYV(:, 002) = -5.600000000000E-002  -5.530000000000E-002  -5.290000000000E-002
      -4.920000000000E-002  -4.480000000000E-002  -4.250000000000E-002
      -3.860000000000E-002  -3.400000000000E-002  -2.680000000000E-002
      -8.900000000000E-003  1.780000000000E-002  2.670000000000E-002
      3.490000000000E-002  3.980000000000E-002  4.230000000000E-002
      4.450000000000E-002  4.850000000000E-002  5.200000000000E-002
      5.470000000000E-002  5.600000000000E-002

```

```

! Keepout Polygon 003

```

```

KOPOLYU(:, 003) = 1.000000000000E+000  9.880000000000E-001  9.732000000000E-001
      9.440000000000E-001  9.151000000000E-001  8.862000000000E-001
      8.718000000000E-001  8.584000000000E-001  8.517000000000E-001
      8.526000000000E-001  8.534000000000E-001  8.544000000000E-001
      8.607000000000E-001  8.765000000000E-001  8.932000000000E-001

```

	9.098000000000E-001	9.432000000000E-001	9.599000000000E-001
	9.932000000000E-001	1.000000000000E+000	
KOPOLYV(:, 003) =	1.041000000000E-001	1.042000000000E-001	1.045000000000E-001
	1.063000000000E-001	1.092000000000E-001	1.127000000000E-001
	1.148000000000E-001	1.214000000000E-001	1.354000000000E-001
	1.524000000000E-001	1.693000000000E-001	1.863000000000E-001
	2.015000000000E-001	2.067000000000E-001	2.090000000000E-001
	2.110000000000E-001	2.141000000000E-001	2.151000000000E-001
	2.165000000000E-001	2.165000000000E-001	

!-----

! Keepout Polygon 004

KOPOLYU(:, 004) =	0.000000000000E+000	1.810000000000E-002	4.880000000000E-002
	8.030000000000E-002	1.124000000000E-001	1.457000000000E-001
	1.627000000000E-001	1.792000000000E-001	1.866000000000E-001
	1.856000000000E-001	1.840000000000E-001	1.831000000000E-001
	1.724000000000E-001	1.563000000000E-001	1.398000000000E-001
	1.068000000000E-001	7.430000000000E-002	4.220000000000E-002
	9.700000000000E-003	0.000000000000E+000	
KOPOLYV(:, 004) =	1.041000000000E-001	1.043000000000E-001	1.055000000000E-001
	1.076000000000E-001	1.103000000000E-001	1.131000000000E-001
	1.150000000000E-001	1.217000000000E-001	1.376000000000E-001
	1.736000000000E-001	1.915000000000E-001	1.915000000000E-001
	2.038000000000E-001	2.077000000000E-001	2.094000000000E-001
	2.125000000000E-001	2.149000000000E-001	2.164000000000E-001
	2.165000000000E-001	2.165000000000E-001	

!-----

! Keepout Polygon 005

KOPOLYU(:, 005) =	1.000000000000E+000	9.888000000000E-001	9.734000000000E-001
	9.423000000000E-001	9.111000000000E-001	8.953000000000E-001
	8.796000000000E-001	8.668000000000E-001	8.650000000000E-001
	8.692000000000E-001	8.732000000000E-001	8.771000000000E-001
	8.812000000000E-001	8.921000000000E-001	9.076000000000E-001
	9.244000000000E-001	9.412000000000E-001	9.580000000000E-001
	9.913000000000E-001	1.000000000000E+000	
KOPOLYV(:, 005) =	2.700000000000E-001	2.701000000000E-001	2.706000000000E-001
	2.720000000000E-001	2.740000000000E-001	2.753000000000E-001
	2.775000000000E-001	2.856000000000E-001	3.002000000000E-001
	3.154000000000E-001	3.306000000000E-001	3.457000000000E-001
	3.607000000000E-001	3.722000000000E-001	3.777000000000E-001
	3.784000000000E-001	3.778000000000E-001	3.773000000000E-001
	3.761000000000E-001	3.760000000000E-001	

!-----

! Keepout Polygon 006

KOPOLYU(:, 006) =	0.000000000000E+000	4.100000000000E-003	3.390000000000E-002
	6.310000000000E-002	9.230000000000E-002	1.221000000000E-001
	1.520000000000E-001	1.667000000000E-001	1.802000000000E-001
	1.876000000000E-001	1.857000000000E-001	1.853000000000E-001
	1.840000000000E-001	1.700000000000E-001	1.536000000000E-001
	1.212000000000E-001	8.880000000000E-002	5.660000000000E-002
	2.420000000000E-002	0.000000000000E+000	
KOPOLYV(:, 006) =	2.698000000000E-001	2.698000000000E-001	2.695000000000E-001
	2.700000000000E-001	2.711000000000E-001	2.728000000000E-001
	2.746000000000E-001	2.761000000000E-001	2.817000000000E-001
	2.943000000000E-001	3.264000000000E-001	3.424000000000E-001
	3.585000000000E-001	3.677000000000E-001	3.699000000000E-001
	3.708000000000E-001	3.721000000000E-001	3.735000000000E-001
	3.748000000000E-001	3.755000000000E-001	

!-----

TARGET_COILPOLY(001) =	0.000000000000E+000
SIGMA_COILPOLY(001) =	1.000000000000E-001
TARGET_COILPOLY(002) =	0.000000000000E+000
SIGMA_COILPOLY(002) =	1.000000000000E-001
TARGET_COILPOLY(003) =	0.000000000000E+000
SIGMA_COILPOLY(003) =	1.000000000000E-001


```
TARGET_COILPOLY( 004) = 0.000000000000E+000
SIGMA_COILPOLY( 004) = 1.000000000000E-001
TARGET_COILPOLY( 005) = 0.000000000000E+000
SIGMA_COILPOLY( 005) = 1.000000000000E-001
TARGET_COILPOLY( 006) = 0.000000000000E+000
SIGMA_COILPOLY( 006) = 1.000000000000E-001
!-----
```

Additionally, the following namelist section should be appended to the end of the STELLOPT input file. This namelist sets parameters for the call to the mgrid generator. The values of RMIN, RMAX, ZMIN and ZMAX should be checked that they are sensible for the particular configuration being studied.

```
&MGRID_NLI
MGRID_MODE = 'S'
LSTELL_SYM = T
RMIN = 6.000000000000E+000
RMAX = 1.400000000000E+001
ZMIN = -4.000000000000E+000
ZMAX = 4.000000000000E+000
KP = 32
JZ = 101
IR = 101
/
```