SIMONS SUMMER SCHOOL COMPUTER LAB PART II: ROTATING ELLIPSE COILS

1. Introduction

REGCOIL is a code which computes stellarator coil shapes. The method is described in detail in the original paper here. One of the inputs to the code is a target equilibrium magnetic surface. The coil shapes are optimized such that the resulting magnetic field lies approximately tangent to this surface. This is characterized by small values of the following function,

$$\chi_B^2 = \int_{S_{\text{plasma}}} d^2x \, (\boldsymbol{B} \cdot \hat{\boldsymbol{n}})^2 \,, \tag{1}$$

where S_{plasma} is the target magnetic surface. For this activity, S_{plasma} will be taken to be the rotating ellipse boundary magnetic surface considered in Part I.

In REGCOIL, an assumption is made that all coils lie on a toroidal winding surface, S_{coil}. For this activity, we will make the assumption that this surface is a uniform offset from S_{plasma}. An example is shown in figure 1: a rotating ellipse boundary ($N_{FP} = 3$, $R_0 = 5$ m, $Z_0 = 0$ m, $a = 2$ m, $b = 1$ m) is shown with a winding surface uniformly offset by 0.8 m. An additional simplifying assumption is made: the number of coils is large such that they can be approximated by a current density on S_{coil}, which we call $\boldsymbol{K}$. After the current density is computed from the REGCOIL solution, individual coils can be obtained with a post-processing tool.

When optimizing stellarator coils it is also necessary to consider how complex their shapes are. An optimal coil set should not have regions of larger curvature, as this is difficult to produce with finite-thickness material. Coils should not be too long, as they take excess material to build. The coils should also not be too close to each other, as it will be difficult to access the confinement region for maintenance

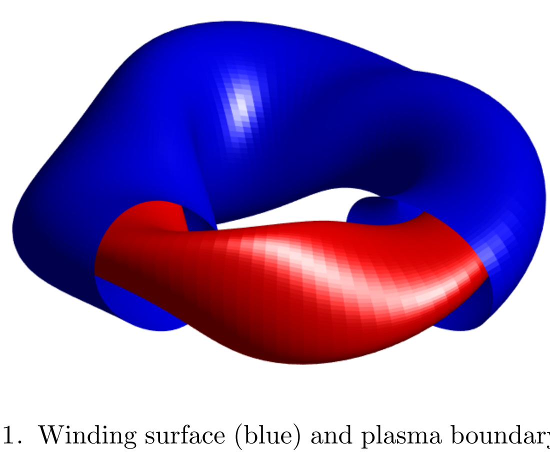

Figure 1. Winding surface (blue) and plasma boundary (red) for rotating ellipse configuration.

and diagnostics. Therefore in addition to (1), we consider a scalar function which quantifies coil complexity,

$$\chi_K^2 = \int_{S_{\text{coil}}} d^2x \, |\boldsymbol{K}|^2. \tag{2}$$

Coils which feature small coil-coil spacing will have larger corresponding values of $|\boldsymbol{K}|$. Thus smaller values of χ_K^2 are correlated with simpler coil shapes.

In REGCOIL the following objective function is minimized to obtain optimal coils,

$$\chi^2 = \chi_B^2 + \lambda \chi_K^2. \tag{3}$$

The regularization parameter, λ, is chosen in order to balance the two competing objectives. This objective function is minimized using a linear least-squares method. Thus the algorithm is guaranteed to obtain the global minimum with only one linear solve. In the following section we review the pertinent input parameters.

2. REGCOIL input parameters

The main input parameters of interest are described below. For more details you can see the user manual here: https://terpconnect.umd.edu/~mattland/regcoilManual.pdf

- `general_option`: controls how λ is chosen in (3). In this activity we will use `general_option` = 5, such that λ is chosen to obtain a certain value of a specified target function. When using this option, `target_option` and `target_value` must be set.
- `target_option`: determines which target function is used in order to determine λ. For this activity we will use `target_option = 'rms_Bnormal'` such that

$$||\boldsymbol{B} \cdot \hat{\boldsymbol{n}}||_2 = \sqrt{\int_{S_{\text{plasma}}} d^2x \, \left(\boldsymbol{B} \cdot \hat{\boldsymbol{n}}\right)^2 / A_{\text{plasma}}} \tag{4}$$

  is our target function, where A_{plasma} is the area of S_{plasma}.
- `target_value`: sets the desired value of the target function used to determine λ. With `target_option = 'rms_Bnormal'`, smaller values of this quantity correspond to more complex coils which reproduce the desired magnetic surface well, while larger values correspond to simpler coils which do not reproduce the desired surface as well.
- `nlambda`: maximum number of values of λ to evaluate. When using `general_option = 5`, a nonlinear root solver is used to obtain the value of λ which corresponds to the desired value of the target function.
- `ntheta_plasma`: number of poloidal gridpoints used to discretize S_{plasma}. Typically 64-128 is sufficient.
- `ntheta_coil`: number of poloidal gridpoints used to discretize S_{coil}. Typically 64-128 is sufficient.
- `nzeta_plasma`: number of toroidal gridpoints used to discretize S_{plasma}. Typically 64-128 is sufficient.
- `nzeta_coil`: number of toroidal gridpoints used to discretize S_{coil}. Typically 64-128 is sufficient.
- `mpol_potential`: number of poloidal Fourier modes used to discretize $\boldsymbol{K}$. Typically 6-12 is sufficient.

- `ntor_potential`: number of toroidal Fourier modes used to discretize $\boldsymbol{K}$. Typically 6-12 is sufficient.
- `geometry_option_plasma`: sets the geometry used to describe S_{plasma}. For this activity we will use `geometry_option_plasma = 2` such that the geometry is read from a VMEC output file.
- `wout_filename`: name of VMEC output file for use with `geometry_option_plasma = 2`. This should include the relative or absolute path.
- `geometry_option_coil`: sets the geometry used to describe S_{coil}. For this activity we will use `geometry_option_coil = 2` such that a uniform offset surface is defined. In this case, `separation` must be specified.
- `separation`: distance from S_{plasma} in meters used to define a uniform offset surface, S_{coil}.

3. Suggested activities

For a complete description on executing the REGCOIL code from the Mac workstations or the PPPL cluster, see the 'Simons summer school computer lab' document.

Both a MATLAB script, `plotCoilsFromRegcoil.m` and a python script, `plotCoilsFromRegcoil.py`, have been included in the `regcoil_inputs` directory. Both of these require specification of the regcoil output file and the number of coils per half period. While MATLAB is not installed on the Mac cluster, either script can be used on the PPPL cluster.

A second python script, `regcoilPlot.py`, has been included. This plots additional quantities that won't be needed for this activity, but those who are curious can use it.

(1) Use the template input file, `regcoil_in.rotating_ellipse` to run REGCOIL using a VMEC output file. An output file obtained during Part I or the included example file, `wout_rotating_ellipse.nc`, can be used.
  - Use the python or MATLAB sciprt to plot the resulting coils. One of the parameters required for this script is the number of coils per half period, `coilsPerHalfPeriod`. Try out a few values (3-7 is usually sufficient).
  - Try adjusting the values of `separation` and `target_value` while running REGCOIL and `coilsPerHalfPeriod` when executing the script. How does the coil complexity compare?

(2) Perform a comparison between rotating ellipse configurations with different values of N_{FP}, keeping all other parameters (R_0, Z_0, a, b) the same. Use the same value of `target_value` and `separation` when running REGCOIL for each of the configurations.
  - Plot the coils for each configuration. For a fair comparison between configurations, use the same total number of coils ($2N_{FP} \times$ `coilsPerHalfPeriod`). How does the coil complexity change as N_{FP} increases?

(3) Perform a comparison between rotating ellipse configurations with different values of R_0, keeping all other parameters the same. Use the same values of `target_value` and `separation` when running REGCOIL for each of the configurations.

- Plot the coils for each configuration. How does the coil complexity change as R_0 increases?

(4) *Challenge activity*: compute the coil metrics (coil-coil distance, toroidal extent, curvature, length) for some of the configurations explored above using the `compareCoilMetrics.m` script, found in the `regcoil_inputs/coilMetricScripts` directory. Which configurations have the "best" coil metrics? Perform the comparison at the same values of `target_value`, `separation` and total number of coils.

If you want to continue playing with regcoil after the summer school lab, the code can be downloaded from https://github.com/landreman/regcoil.