**Rayan Sud  
11/09/2019  
ITER**

**IAIA Reconstruction**

**User Guide**

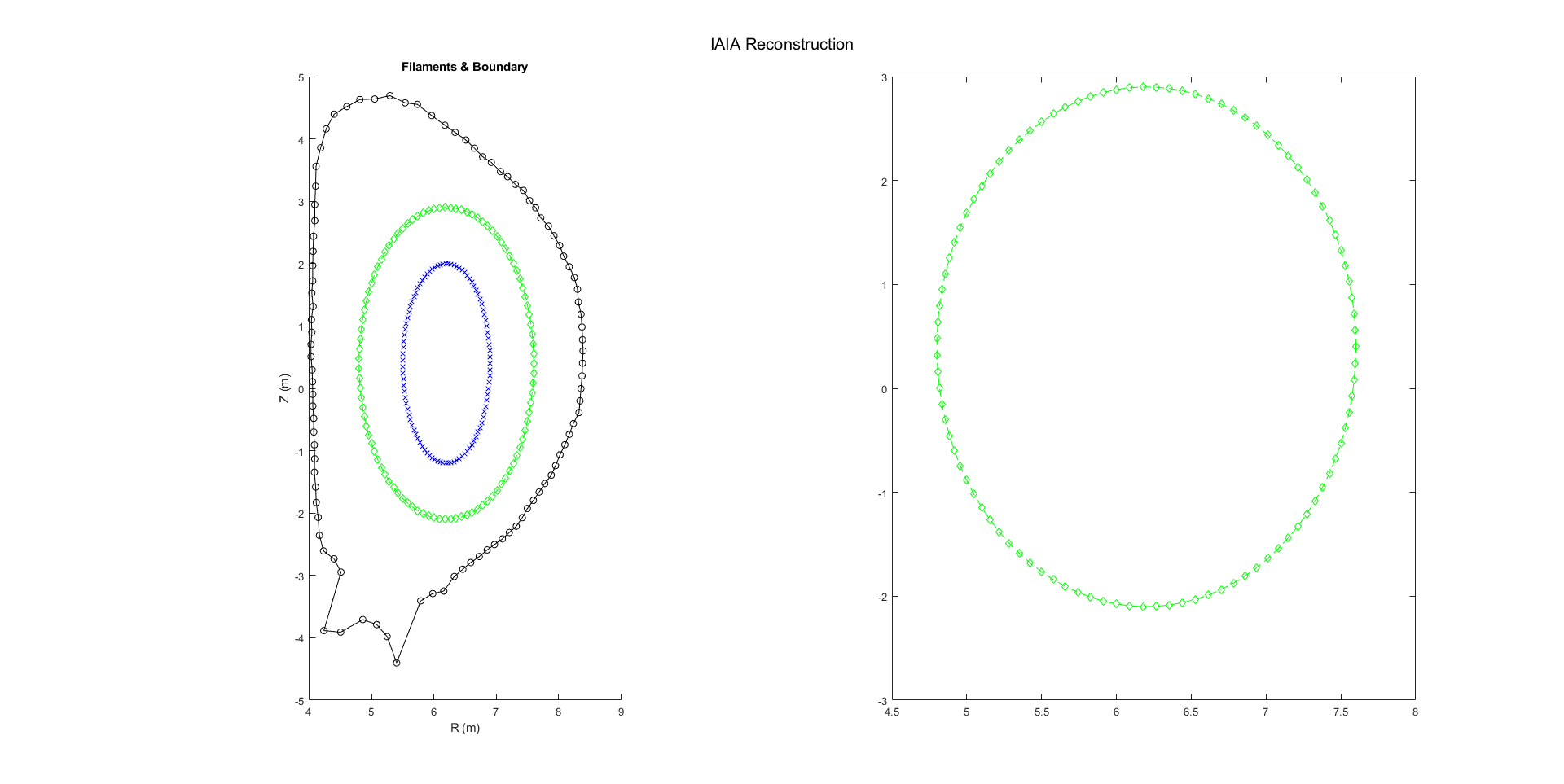
Based on [this](https://www.sciencedirect.com/science/article/pii/S0920379612005340?via%3Dihub) paper by Bettini, I describe below a process to find the boundary of an ITER plasma.

**Program Description:**

For a full explanation of this program, see the above linked paper, as well as Rayan Sud’s Internship Report, Section 3, [IDM LINK].

In brief, this program uses magnetic diagnostic information to reconstruct a plasma boundary. It uses a set of filaments to approximate the plasma, and iteratively moves them outwards until convergence. The plasma current is found using a Green matrix inversion by truncated Singular Values Decomposition.

**User Guide:**

1. Change the parameters and input data files to your desired scenario.
   1. SourceCoils.csv  
      This CSV file must contain all sources whose effect you wish to remove from the reconstruction. This allows the program to isolate the effect of the plasma. The CSV may contain an arbitrary number of sources, including PF & CS coils, or wall currents. It must contain the R-Z coordinates of the sources in meters, and the current per turn and number of turns. The headers should be: R, Z, I, n.
   2. MagneticData.csv  
      This CSV file must contain the R-Z coordinates in meters, angular orientation, and measured B field of all magnetic sensors in T. The headers must be: R, Z, Main Axis Poloidal Angle, B.
   3. LimiterLeftBoundaryWall.csv  
      This file describes the coordinates of the inner wall of the vessel. The program uses these points to search for the limiter point, in limiter configuration.
   4. Geometry.mat  
      This file contains the coordinates of many geometrical shapes. All are column vectors, in meters.
      1. *R\_EC0, Z\_EC0*  
         These variables describe the positions of the inner layer of Equivalent Currents (ECs). These should be well inside any plasma boundary. (shown in blue crosses)
      2. *R\_wall, Z\_wall*  
         These variables contain a subset of ITER first wall coordinates. The coordinates are synchronized with the EC coordinates, such that they are roughly evenly distributed, and line up with their correspondingly indexed EC coordinates. This synchronization is mandatory for the proper operation of the program (shown in black circles). The lengths of these vectors must equal the lengths of the R\_EC0, Z\_EC0, R\_guess, Z\_guess vectors.
      3. *R\_guess, Z\_guess*  
         These variables contain the coordinates of the initial guessed plasma boundary. As with the R\_wall, Z\_wall coordinates, they must be synchronised with the R\_EC0, Z\_EC0 coordinates (shown in green).
      4. *R\_realplasma, Z\_realplasma*  
         These describe the real boundary of the plasma you are trying to approximate. This is used only for testing – these variables could be replaced with scalars of value 0 if the program is actually being used.
      5. *R\_fullwall, Z\_fullwall*  
         These variables describe the coordinates of the ITER first wall. These variables are used only for aesthetic purposes, in plotting.
   5. N\_ITERATIONS  
      This variable is present in the IAIA.m file. It is set to 3 by default. Increasing or decreasing this number will change the number of iterations the program performs to find the boundary.
   6. XP\_RMIN, XP\_RMAX, XP\_ZMIN, ZP\_ZMAX  
      These coordinates define the bounding coordinates for the box where the program searches for an X-Point. These can be altered if the X-point is to be found in a different region
   7. GRID\_RES  
      This constant sets the resolution of points
   8. Limiter, xpoint  
      These variables are essentially Booleans. Setting limiter=1, xpoint=0 makes the program to limiter configuration, and similarly limiter=0,xpoint=1 sets it to x-point configuration.
   9. TRUNCATION\_PERCENTAGE (inside solveForIp function)  
      This variable manages the truncation index of the TSVD process used to solve for the EC current distribution. Any singular values below this fraction of the maximum value are discarded.
   10. ERROR\_TOLERANCE, STEP\_SIZE (inside findXPointFlux function)  
       These constants manage the gradient descent x-point finding algorithm. The error tolerance is the distance, such that if an iteration of gradient descent moves to a point less than this distance away from its previous location, the algorithm terminates and returns the last point found. The step size is the fixed step length that the algorithm takes in the direction of a saddle point.
2. Run “IAIA.m”.