Use of Radial Basis Functions for Fixed Boundary Equilibrium Reconstruction

E. Giovannozzi[1], G. Calabrò[1], F. Crisanti[1], P. Micozzi[1]
and EFDA ITM-TF contributors[2]
[1]*Associazione EURATOM - ENEA sulla fusione, Frascati, Italy*
[2]*http://www.efda-itm.eu*

Introduction

Radial basis functions (RBF) have been used for solving elliptic problems [1,2]. We present an application of them to solve fixed boundary tokamak equilibria. For the description of the problem we follow the EU Integrated Tokamak Modelling Task Force (ITM-TF) conventions. The equation to solve is:

$$\frac{1}{2\pi}\,\Delta^*\Psi = -2\pi\left(\mu_0 R^2\frac{\partial p}{\partial\Psi} + F_{dia}\frac{\partial F_{dia}}{\partial\Psi}\right) = \mu_0 R\, j_\phi \tag{1}$$

Where Δ^* is the usual Grad-Shafranov operator $\Delta^* = R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial}{\partial R}\right) + \frac{\partial^2}{\partial Z^2}$, Ψ is the poloidal flux function, R and Z are cylindrical coordinates and p and F_{dia} are two arbitrary functions depending only on Ψ. In the RBF approximation a function is approximated by a sum of radial functions centred on N different nodes $\Psi(R,Z) = \sum_{i=1}^{N} w_i\phi(r_i,d_0)$, with $r_i = \sqrt{\left(R - R_i\right)^2 + (Z - Z_i)^2}$ where (R_i, Z_i) is the position of each node, and d_0 a scale length. The Grad-Shafranov equation can be solved substituting the expansion of Ψ in terms of radial basis functions in equation (1). In order to calculate the N weights w_i we need to solve N algebraic equations. Some points are chosen in the integration domain, the points will be divided in N_{int} internal, where the GS equation is imposed and N_{bound} boundary points ($N_{int} + N_{bound} = N$), where the boundary condition ($\Psi\left(\vec{x}_{i,bound}\right) = 0$) is valid, generally the chosen points are coincident with the nodes. So we have N_{int} equations:

$$\Delta^*\Psi(\mathrm{R_k}, \mathrm{Z_k}) = \sum_{i=1}^{N} w_i\left(\frac{R_i}{r_{ki}R_k}\frac{d\phi(r_{ki},d_0)}{dr} + \frac{d^2\phi(r_{ki},d_0)}{dr^2}\right) = \left(\mu_0 R j_{phi}\right)\big|_{R,Z=R_k,Z_k} \tag{2}$$

and N_{bound} equations $\sum_{i=1}^{N} w_i\phi(r_{ji},d_0) = 0$. The problem then reduces to solving a system of equation for the unknown weights: $\boldsymbol{Aw} = \boldsymbol{b}$. Generally the matrix **A** is not sparse and it can also become more ill conditioned when increasing d_0, which generally increase the precision of the solution. Methods to overcome the ill conditioning of the matrix have been developed [3] they will be tested in a future work.

The widely used radial basis function is $\phi(r, d_0) = \sqrt{\frac{r^2}{d_0^2} + 1}$ has been adopted. The node positions has been chosen using a principle of minimal energy, basically we minimize a global potential $W = \sum_{i,j} \frac{\exp(-\alpha r_{i,j})}{\sqrt{r_{i,j}^2 + l^2}}$ where $r_{i,j} = \|\vec{x}_i - \vec{x}_j\|$ is the distance between two nodes. The parameter l is used in order to avoid a divergence for $r_{i,j} = 0$ while the parameter α can be used to avoid long range interaction between points and have a more uniform distribution. The nodes on the boundary are kept fixed while only the internal nodes are free to move during the optimization.

In general $\partial p/\partial\Psi$ (usually called p') and $F_{dia}\,\partial F_{dia}/\partial\Psi$ ($F_{dia}F'_{dia}$) depend on Ψ, so the solution should be obtained by iterations. As it is usual the problem is restated as:

$$\frac{1}{2\pi}\Delta^*\Psi^*_{k+1} = -2\pi\left(\mu_0 R^2 \frac{\partial p}{\partial\bar{\Psi}_k} + F_{dia}\frac{\partial F_{dia}}{\partial\bar{\Psi}_k}\right)$$

Where $\bar{\Psi}_k = \frac{\Psi^*_k}{\Psi^*_{Mk}}$ is the normalized flux function at the previous step, and Ψ^*_{Mk} is the maximum of the flux function at the iteration k, basically the factor needed to get a normalized flux function. The solution to the original equation is then $\Psi = \frac{\Psi^*}{\sqrt{\Psi^*_M}}$.

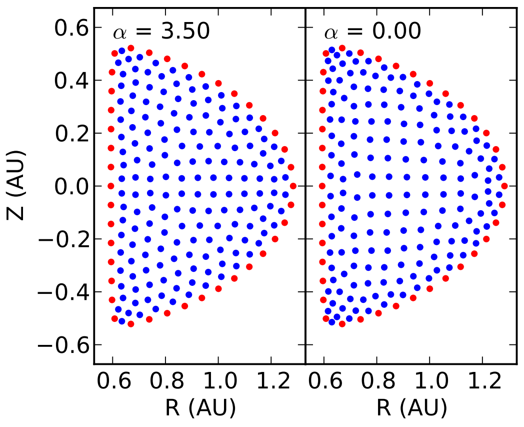

Fig. 1. Grid point positions obtained with different value of the parameter α. The total number of grid points is 206. With 40 points on the boundary (red).

In a typical fixed boundary code additional constraints are imposed on the total current and the poloidal beta, but in the present situation, where we compare our output to an already calculated equilibrium, they are not needed as long as we use the correct p' and $F_{dia}F'_{dia}$. The code has been written in Python, which provides an extensive set of function for optimization and graphics, and implemented on the ITM Gateway. The infrastructure there available permits an easy and standard interface to the output of other equilibrium codes.

Soloveev solution

The code has been tested against the Soloveev solution [4], one possible parameterization of which is the following one where $\frac{\partial p}{\partial\Psi}$ and $F_{dia}\frac{\partial F_{dia}}{d\Psi}$ are constant and don't depend on Ψ:

$$\Psi(r,z) = \Psi_0 - D^2(r^2 - r_0^2)^2 - \frac{1}{2}\left(\left(4\pi^2\mu_0\frac{\partial p}{\partial\Psi} - 8D^2\right)r^2 + 4\pi^2 F_{dia}\frac{\partial F_{dia}}{d\Psi}\right)z^2 \qquad (3)$$

In the equation D is an additional constant and r_0 is the position of the magnetic axis. As the right hand side of the equations (**b**) doesn't depend on Ψ the solution converges in just one iteration. Generally for larger d_0 the solution improves up to the point where the ill conditioning prevents further increasing of the precision. We checked the solver against a Soloveev equilibrium with the following parameters: $r_0 = 1.0\,m$, $\mu_0 p' = 0.32\ T/m^2$, $F_{dia}F'_{dia} = 0.057\,T$, $D = 0.76\,Wb$, $\Psi_0 = 0.32\,Wb$). In the following figures we calculated the maximum difference between the calculated solution (Ψ_{solv}) and the reference solution (Ψ_{ref}):

$$\Delta\Psi = \max(\Psi_{solv} - \Psi_{ref})/\max\Psi_{ref}$$

Figure 2 illustrates the effect of different position of the nodes on the precision of the obtained solution. Not a big difference is seen between the two cases, even though it not excluded that a better positioning of the points can get a better precision.

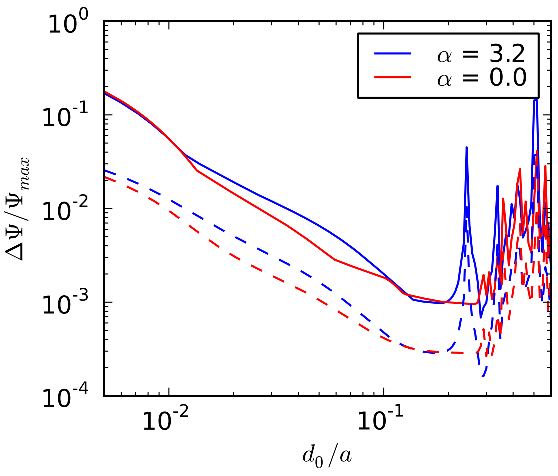

Fig. 2 Effect of the grid points distribution on the precision of the solver. The continuous lines are the maximum differences between the calculated and analytical one, while the dashed line is the root mean square deviation between the two.

The continuous lines refer to the maximum deviation from the analytical one relative to the maximum value of the Ψ while the dashed one refers to the relative square root deviation. A global scale factor close to one is still present. For large d_0 the sharp increase to the deviation are likely due to the ill conditioning of the matrix **A**. The Solovev solution can be used to test the limits of this kind of solver, in figure 3 it is shown the effect of different number of points as function of the parameter d_0/a, where a is the plasma minor radius. There is a limit on the precision of about $\sim 10^{-3}$, this limit is obtained for $d_0/a \sim 0.3$, and $N_{grid} > 100$.

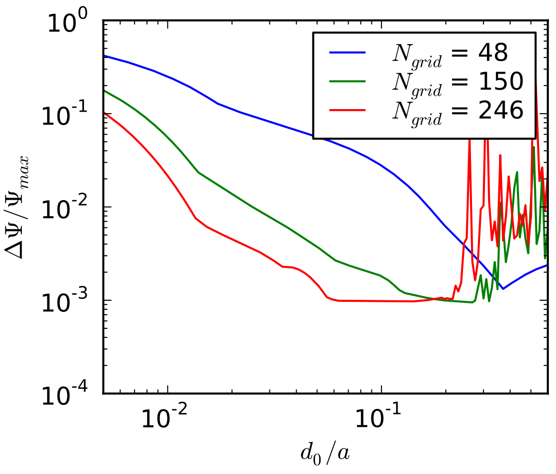

Fig. 3 Effects of the number of point on the precision of the solution. While there is an increase of precision from 48 grid points and 150 grid points, not much is gained from going from 150 to 246 points.

Comparison to FIXFREE

FIXFREE is a free boundary equilibrium code based on the expansion of the GS solution in toroidal harmonic series [5]. The comparison has been performed within the ITM-TF simulation framework, where standard inputs and outputs are defined. A solution of the GS corresponding to an equilibrium of FAST [6] has been calculated by FIXFREE. We take from the output the position of the boundary and the p' and $F_{dia}F'_{dia}$. The RBF solver is then applied, the resulting Ψ is then compared with the output of FixFree. As the right hand side of the GS equation depends on the calculated value of the flux-function the solution is obtained only after some iterations. The iterations stopped when the weights w_i change less then 10^{-6} compared to the maximum value of the weights. This is basically achieved in ~15 iterations. The calculated solution by the RBF solver agrees with the solution coming from FIXFREE with a precision of the order of 10^{-2}, this is obtained for a $d_0/a \approx 0.2$ A slightly better result is obtained with a grid of only 47 points, but further investigations are necessary.

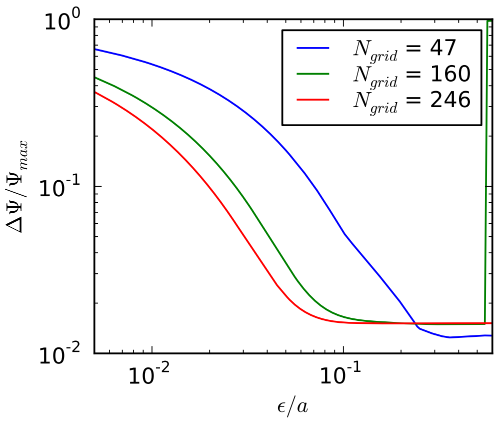

Fig. 4 Comparison between this solver and FixFree. As before 160 points gives the best performance.

Conclusions

A fixed boundary equilibrium solver has been developed, based on a radial basis function expansion of the flux-function. A precision of at least 10^{-3} has been obtained in the comparison against a Soloveev solution, while an agreement of about 10^{-2} has been obtained against FIXFREE code. Realization of the solver in Python language provides an extensive set of function for optimization and graphics. The implementation inside the ITM-TF framework permits the easy comparison of the results with other equilibrium solvers. This work is in progress as well as determination of the precision limits of the code.

Acknowledgments

This work, supported by the European Communities under the contract of Association between EURATOM-ENEA was partly carried out within the framework of the Task Force on Integrated Tokamak Modelling of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] E.J. Kansa, Computers & Mathematics with Applications, 19, 8–9, p 127-145 (1990)
[2] E.J. Kansa, Computers & Mathematics with Applications, 19, 8–9, p 147-161 (1990)
[3] L. Ling, E.J. Kansa, Advances in Computational Mathematics 23: 31–54 (2005)
[4] L. S. Solov'ev, in Reviews of Plasma Physics, edited by M. A. Leontovich Consultants Bureau, New York, 6, p. 239 (1976)
[5] F. Alladio, F. Crisanti, M. Marinucci, P. Micozzi, A. Tanga, Nucl. Fusion 31, 4, p 739 (1991)
[6] A. Pizzuto et al 2010 Nucl. Fusion 50, 9, 095005 (2010)