Full discharge coil trajectory optimisation using a quasi-Newton method with the FBT code from the MEQ suite

A. Merle[1], F. Felici[2], C. Heiss[1], G. Van Parys[1], J. Wai[3]

[1] *École Polytechnique Fédérale de Lausanne, Swiss Plasma Center, Lausanne, Switzerland*

[2] *Google DeepMind, London, UK*

[3] *Commonwealth Fusion Systems, Devens, Massachusetts 01434, USA*

Introduction

Optimising coil current trajectories is an essential task in the preparation of a tokamak discharge. It can help ensure that the scientific objectives are fulfilled in terms of main plasma separatrix position and shaping as well as secondary x-point and strike point positions while minimising the total amount of current run through the poloidal field coils and the flux consumption in the transformer and respecting the operational boundaries of the machine, coils and associated power supplies.

The MEQ suite

The MEQ (Matlab EQuilibrium) suite is comprised of fast and reliable codes to compute the axisymmetric configuration of the magnetic field in a tokamak. The codes provide free-boundary solutions to the ideal MHD Grad-Shafranov equation. For the evolutive problem the Grad-Shafranov equation is coupled to circuit equations for active and passive toroidal conductors and different forms of the current diffusion equation can also be included. The solvers use a Cartesian grid in the (R,Z) plane and the plasma profiles p' and TT' are described as the sum of a predefined set of basis functions. The codes can also treat multiple plasma domain configurations such as doublets or droplets, and an extension of the model to include toroidal flow and pressure anisotropy is being implemented. The suite contains both forward solvers and optimisation solvers. FGS and FGE are forward solvers for the static and evolutive problems respectively [1], LIU is an optmisation solver for the equilibrium reconstruction problem [3] and FBT is an optimisation solver for the inverse equilibrium problem [2].

FBT and the inverse equilibrium problem

The FBT code [2] solves an optimisation problem to obtain coil trajectories based on user constraints. The cost function is constructed in a linear least squares fashion with terms corresponding to flux values (absolute or relative to some unknown F_b value), radial or vertical components of the poloidal field, orientation of the poloidal field, flux hessian values and vacuum poloidal field gradients all defined at a set of given control points. Additionally some regularisation constraints can be added based on individual coil current values I_a or coil current dipole values or some user-defined coil current combinations. All cost constraints can also be added as exact constraints. p' and TT' profiles are determined using as many exact constraints as basis function coefficients. Inequality constraints can also be added to ensure that the solution stays within the allowed operational range based on individual coil current values or combinations thereof.

FBT has been extended to treat the evolutive problem where circuit equations are included in the model equations coupling time slices together [4, 5]. Additional optimisation objectives (cost, equality or inequality constraints) include vessel current values I_u and power supply voltages V_a. The objectives can also include the first or second order time-derivatives of the quantities described previously. Finally relative weights of objectives for individual equilibria can be introduced.

Optimising equilibria

Given a cost function $W(x,u,z)$, we are looking for the solution (x,u,z) that minimises W such that the model equations $F(x,u) = 0$ are also verified. Here x represents the system state, in particular $J_x = \partial F/\partial x$ is a square full-rank matrix, u represents the system inputs corresponding to quantities other than the system state entering the model equations and z represents the optimisation variables that do not enter the model equations (e.g. boundary flux, lagrange multipliers, stabilisation parameter ...). In the evolutive problem x represents the collection of system states for all time slices, u and z are similarly expanded.

Previously FBT used an algorithm based on Picard iterations. Schematically it would use a 2 stage optimisation process where first the cost function is minimised against u and z at fixed x and x is then updated at fixed u towards the $F(x,u) = 0$ constraint. At convergence the solution verifies

$$\frac{\partial W}{\partial u} = 0, \quad \frac{\partial W}{\partial z} = 0, \quad F(x,u) = 0 \tag{1}$$

A new algorithm has been implemented based on a Quasi-Newton Sequential Quadratic Programming method, which was previously implemented in the NICE code [6].

Given (x_n, u_n, z_n) the next iteration is obtained using

$$\min_{du,dz} W(x_n + dx_{0,n} + dx/du \cdot du, u_n + du, z_n + dz) \tag{2}$$

with $dx_{0,n} = -J_x^{-1} F(x_n, u_n)$ and $dx/du = -J_x^{-1} \cdot J_u$. It is a Quasi-Newton method since terms corresponding to the second derivatives of F have been dropped. At convergence the solution verifies

$$\frac{\partial W}{\partial x}\frac{dx}{du}+\frac{\partial W}{\partial u}=0, \quad \frac{\partial W}{\partial z}=0, \quad F(x,u)=0 \tag{3}$$

It can be shown that Picard iterations are Newton iterations with approximate J_x, J_u such that $\partial\{j_\phi, C_o\}/\partial\Psi = 0$.

Implementation and verification in MEQ

MEQ makes use of a fully analytic computation of the jacobian of the model equations. Each iteration requires solving a linear system $J_x \mathrm{d}x = b$ with multiple RHS. These are solved using either LU decomposition of J_x or a Block-GMRES method. The reduced quadratic optimisation problem is solved using an interior point method. One useful feature of the set of FBT optimisation objectives described earlier is that they are all linear in x, u, z.

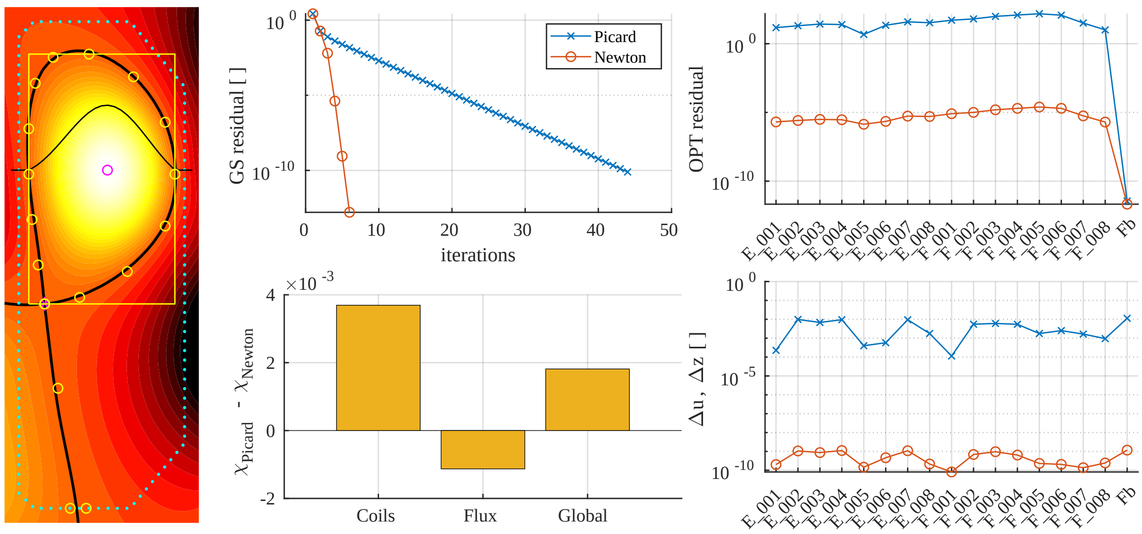

Figure 1: Comparison of Picard and Quasi-Newton methods for a static TCV equilibrium

Figure 1 shows the result of an FBT run for a static TCV diverted case. The Quasi-Newton method provides quadratic convergence which is faster than the Picard method, but each iteration is more expensive yielding overall similar computational time. Despite the approximations present in the Picard method, the optimised coil currents and resulting equilibria are often quite similar in most cases.

Figure 2 show the results of the optimisation of a TCV ramp-up scenario without including the ohmic transformer circuit or a current diffusion equation. This new FBT version was also used to design feed-forward coil trajectories for the breakdown phase of TCV scenarios for single-axis and doublet plasmas. Since the model is purely linear in the phase with no plasma, no iterations are required.

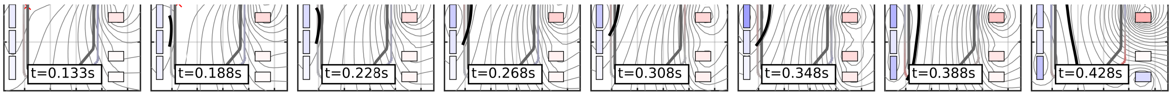

Figure 2: Optimisation of a TCV ramp-up scenario in FBT using a Quasi-Newton method

Conclusions

The implementation of a Quasi-Newton method in FBT provides solutions closer to the optimum than Picard iterations using fewer iterations yet similar run time. This method has been extended to the evolutive problem and allows optimisation of full-discharge coil currents trajectories. Although not shown here this method has also been applied to the equilibrium reconstruction problem (LIUQE) and has shown similar improvements and alleviated the need for a vertical stabilisation scheme. Future developments of FBT will focus on the inclusion of intermediate time steps to help better resolve the vessel currents dynamics and the of a current diffusion equation for the plasma and of transformer circuits in the optimisation. Additionally analytical jacobians will be used to yield estimates of the reconstruction uncertainties in LIUQE.

Acknowledgements

This work was supported in part by the Swiss National Science Foundation. This work has been carried out within the framework of the EUROfusion Consortium, via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion) and funded by the Swiss State Secretariat for Education, Research and Innovation (SERI). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, the European Commission, or SERI. Neither the European Union nor the European Commission nor SERI can be held responsible for them.

References

[1] F. Carpanese, Ph.D. thesis, EPFL (2021)
[2] F. Hofmann *et al.*, Computer Physics Communications **48** (1988) 207
[3] J.M. Moret *et al.*, Fusion Engineering and Design **91** (2015) 1
[4] F. Felici *et al.*, 49th EPS conference on Plasma Physics (2023)
[5] J. Wai, *et al.*, arXiv:2306.13163 (2023)
[6] B. Faugeras *et al.*, Fusion Engineering and Design **160** (2020) 112020