Automating the Design of Tokamak Experiment Scenarios

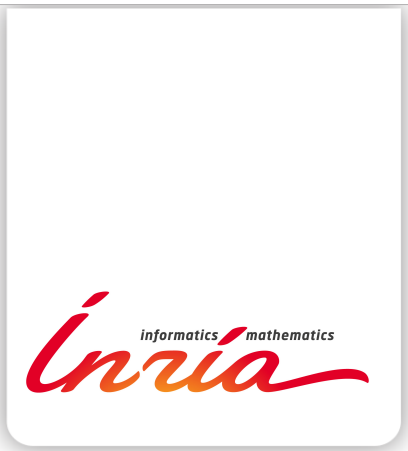
Jacques Blum, Holger Heumann, Eric Nardon, Xiao Song

Automating the Design of Tokamak Experiment Scenarios

Jacques Blum, Holger Heumann, Eric Nardon, Xiao Song

RESEARCH REPORT
N° 9237
December 2018
Project-Teams CASTOR

ISSN 0249-6399 ISRN INRIA/RR--9237--FR+ENG



Automating the Design of Tokamak Experiment Scenarios

Jacques Blum, Holger Heumann, Eric Nardon, Xiao Song

Project-Teams CASTOR

Research Report n° 9237 — December 2018 — 28 pages

Abstract: The real-time control of plasma position, shape and current in a tokamak has to be ensured by a number of electrical circuits consisting of voltage suppliers and axisymmetric coils. Finding good target voltages/currents for the control systems is a very laborious, non-trivial task due to non-linear effects of plasma evolution. We introduce here an optimal control formulation to tackle this task and present in detail the main ingredients for finding numerical solutions: the finite element discretization, accurate linearizations and Sequential Quadratic Programming. Case studies for the tokamaks WEST and HL-2M highlight the flexibility and broad scope of the proposed optimal control formulation.

Key-words: optimal control, PDE-constrained optimization, plasma equilibrium, tokamak

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Automatisation de la conception des scénarios d'expériences Tokamak

Résumé : Le contrôle en temps réel de la position, de la forme et du courant du plasma dans un tokamak doit être assuré par un certain nombre de circuits électriques composés de fournisseurs de tension et de bobines axisymétriques. Trouver de bonnes tensions / courants cibles pour les systèmes de commande est une tâche très laborieuse et non triviale en raison des effets non linéaires de l'évolution du plasma. Nous présentons ici une formulation de contrôle optimal pour aborder cette tâche et présentons en détail les principaux ingrédients permettant de trouver des solutions numériques: la discrétisation par éléments finis, des linéarisations précises et la programmation séquentielle quadratique. Des études de cas sur les tokamaks WEST et HL-2M soulignent la flexibilité et le large champ d'application de la formulation de contrôle optimal proposée.

Mots-clés : contrôle optimal, optimisation sous contrainte EDP, équilibre du plasma, tokamak

AUTOMATING THE DESIGN OF TOKAMAK EXPERIMENT SCENARIOS

JACQUES BLUM, HOLGER HEUMANN, ERIC NARDON, AND XIAO SONG

Abstract. The real-time control of plasma position, shape and current in a tokamak has to be ensured by a number of electrical circuits consisting of voltage suppliers and axisymmetric coils. Finding good target voltages/currents for the control systems is a very laborious, non-trivial task due to non-linear effects of plasma evolution. We introduce here an optimal control formulation to tackle this task and present in detail the main ingredients for finding numerical solutions: the finite element discretization, accurate linearizations and Sequential Quadratic Programming. Case studies for the tokamaks WEST and HL-2M highlight the flexibility and broad scope of the proposed optimal control formulation.

1. Introduction

Nuclear fusion is a highly exothermic reaction in which two light atomic nuclei fuse to form a heavier nucleus. The peaceful use of such reactions for energy production on earth is a multinational research effort with high impact on the long-term perspective of energy production and consumption. The most promising technology to achieve this goal is currently the tokamak, a torus shaped reactor that uses strong magnetic fields to confine plasma and to achieve the extreme conditions to start the fusion reaction.

The mathematical modeling of nuclear fusion physics in such complicated devices is very diverse (see [17] for an detailed introduction, or [6] and [16] for a recent overview). We have an extreme range of spatial and temporal scales. Spatial scales range from 10^{-5} meters for the electron orbit radius to 10^{2} meters which is the length of the magnetic field lines in between the reactor walls. The temporal scales range from 10^{-11} seconds for one period of the electron orbit to 10^{2} seconds for the resistive diffusion time, that describes plasma evolution at the timescale of resistive diffusion. We are dealing here with multi-scale problems and strong anisotropy. So, the modeling in nuclear fusion ranges from high dimensional kinetic models to completely different kind of fluid models with various levels of model reductions steps that reduce the high dimensionality but increase the complexity of the models. The reduced-order models are very non-standard and include to a very high degree empirical intuition.

Nevertheless, the operation of thermonuclear fusion experiments relies heavily on such reduced-order models as first principles approaches are far too expensive to address the many interesting and important questions in tokamak engineering: identification of plasma characteristics in real-time, fast pre-shot simulations within virtual tokamak frameworks, fast post-shot simulations within the same framework augmented by measurement data, development of experimental scenarios, design of

Date: December 18, 2018.

new devices, improvement of heat load mitigation or assessing novel strategies to improve stability, to name just a few.

In this work we will give a contribution to the development of experimental scenarios. Besides many others an experimental scenario in a tokamak requires prescribed time series of voltages applied in the so-called *poloidal field coils* that have an axisymmetric geometry and control form and shape of the plasma inside the tokamak. The basis for designing such voltage time series is the numerical solution of the *free-boundary equilibrium evolution problem* (FBEE). The FBEE problem, as it will be detailed in Section 2, combines the classical Grad-Shafranov equation, describing the magnetohydrodynamic equilibrium in axisymmetric systems, with circuit and induction equations for the poloidal field coils and other passive structures inside the tokamak. For a given time series of voltages $\vec{V}(t) = (V_1(t) \dots V_n(t))$ of n suppliers it allows to predict the evolution the plasma and to check whether a specific choice of voltages can ensure a certain scenario. For example, it can be checked if the voltages enable the transition from limiter to divertor configuration and how much current is accumulated in passive structures and coils, and whether this violates certain technological constraints or not. Hence, the design of voltage trajectories can be regarded as the inverse FBEE problem: We have a certain desired evolution of the plasma in a tokamak in mind and ask to find the voltage trajectories that ensure this. Solving inverse problems is challenging in general. Here, the choice of correct voltages in the FBEE problem is by no means trivial, and can be a very laborious task. The huge variety of realizations of voltage times series and the non-linear nature of the problem make it very difficult to guess the correct voltages. Moreover, a plain trial and error approach is not possible as the plasma equilibria of interest are often unstable, and small variations of the voltages $\vec{V}(t)$ can lead to entirely different results.

We therefore introduce an optimal control approach to scenario design. As the primal unknown of the FBEE problem is the *poloidal magnetic flux* ψ we encode the design goal in an objective functional $C[\psi(t), \vec{V}(t)]$ that is large when the design goal is violated and small otherwise. We define a *convex regularization functional* $R[\vec{V}(t)]$ and formulate a constrained optimization problem

$$\min_{\psi(t), \vec{V}(t)} C[\psi(t), \vec{V}(t)] + R[\vec{V}(t)] \quad \text{subject to } \psi(t), \vec{V}(t) \text{ verify the FBEE problem (see Section 2).} \tag{1}$$

Candidates for objective and regularization functionals are

$$C[\psi(t), \vec{V}(t)] = \frac{1}{2} \int_0^T w(t) \sum_{i=1}^{N_{\text{desi}}} \big(\psi(r_i(t), z_i(t), t) - \psi(r_0(t), z_0(t), t)\big)^2 dt \tag{2}$$

and

$$R[\vec{V}(t)] = \frac{1}{2} \int_0^T \sum_{j=1}^{n} \sum_{i=1}^{n} \mathbf{D}_{ij} V_i(t) V_j(t) dt\,, \tag{3}$$

with the weights $w(t)$ non-negative and $\mathbf{D}$ positive definite. This choice of the objective functional forces $\psi(r, z, t)$ to be constant on a prescribed set of $N_{\text{desi}} + 1$ points $(r_i(t), z_i(t))$ at each instant t and hence can be used to encode a certain desired evolution of the plasma. Other choices of objective functionals could include penalization of induced currents, voltages in suppliers or loop voltages or any other

design goal that can be quantified in terms of the poloidal flux ψ and the voltages $V_i(t)$. The regularization term $R[V(t)]$ enables to work with a well-posed problem, i.e. a problem that is stable to perturbations on the data.

The optimal control formulation (1) alone is only the basis for a powerful tool for scenario design. To put this into practice it is of paramount importance to specify appropriate discretization methods, leading to non-linear algebraic finite-dimensional problems, that are then solved by appropriate iteration schemes. Simple gradient descent (see Appendix A), for example, will suffer from the same difficulties as a trial-and-error approach, as it requires to solve repetitively the non-linear FBEE problem for a varying choice of voltages $V_i(t)$. For that reason, a sequential quadratic programming (SQP) approach (see Appendix A) is here the most appropriate choice. Moreover, it is important to formulate SQP for the discretized version of (1) in order to avoid convergence problems due to inaccurate derivatives. The discretization of (1) and the SQP formulation are outlined in Sections 3.1-3.3.

This work does not claim that (2) and (3) are the exclusive and optimal choices for scenario design but should highlight how different choices of objective functions in (1) can be useful to tackle different tasks in scenario design. We are convinced that robust and easily accessible implementations of (1) will become invaluable tools for designing and improving control for tokamaks. The *optimal* solutions $\psi_*(t)$ and $\vec{V}_*(t)$ of the optimal control problem (1) are not only solutions of the FBEE, but for a reasonable choice of the objective functional, the solutions are also close to the design goals. So, if these design goals strongly penalize unstable behavior (e.g. vertical displacement event, strong induced currents), then we have found a voltage times series $\vec{V}_*(t)$ that provides a plasma equilibrium evolution, even though the equilibrium might pass through physical unstable states. So with this optimal control approach (1) we can separate the problem of finding good feedforward control from the problem of designing effective feedback control as we do not need to include feedback control into our simulation tool to avoid physical instabilities during the simulation. We will illustrate this line of thought with extensive numerical examples in detail in Section 5.

Even though we can get in principle solutions for any kind of objective functional $C[\psi(t), \vec{V}(t)]$, we would like to stress that the whole power of the approach is hidden in an appropriate choice of the objective functional: an objective functional encoding two competing goals will lead to disappointing solutions and an objective functional not penalizing unstable behavior can lead to unstable optimal solutions. But clearly, deciding whether a cost functional encodes competing goals or whether it penalizes sufficiently strong unstable behavior is not easy and requires a good knowledge in both the physical background and the mathematics of optimal control. Optimal control is not an auto pilot, but is capable to leverage considerably the tokamak physicists intuition.

To illustrate the power of the proposed optimal control approach to scenario design we developed an extension of the MATLAB/Octave library FEEQS.M. This library is based on the methods for axisymmetric free boundary plasma equilibria that where described in [3] and [10] and utilizes in large parts vectorization. Therefore, the running time is comparable to C/C++ implementations. FEEQS.M is publicly available[1] and a forthcoming release will contain the methods introduced

[1] http://www-sop.inria.fr/members/Holger.Heumann/Software.html

in this work. The Section 4 contains validation and verification results for these new developments and the Section 5 presents applications for the tokamaks WEST [5] in Cadarache, France and HL-2M [13] in Chengdu, China. We can also refer the related [14], where the same approach was used to identify for WEST configurations with regions of low magnetic fields as it is required for a successful start-up.

2. Free-Boundary Plasma Equilibrium Evolution

The essential equations for describing plasma equilibrium in a tokamak are force balance, the solenoidal condition and Ampère's law that read respectively

$$\operatorname{grad} p = \mathbf{J} \times \mathbf{B}\,, \quad \operatorname{div} \mathbf{B} = 0\,, \quad \operatorname{curl} \frac{1}{\mu}\mathbf{B} = \mathbf{J}\,, \tag{4}$$

where p is the plasma kinetic pressure, $\mathbf{B}$ is the magnetic field, $\mathbf{J}$ is the current density and μ the magnetic permeability. In the quasi-static approximation these equations are augmented by Faraday's law in all other conducting structures, and by Ohm's laws in coils and passive structures. Hence for the resistive timescale the plasma is in equilibrium and (4) holds at each instant of time [8].

Under the assumption of perfect axial symmetry, it is convenient to put (4) in a cylindrical coordinate system (r, φ, z) and to consider only a meridian section of the tokamak. The primal unknowns are the *poloidal magnetic flux* $\psi = \psi(r, z)$, the pressure $p = p(\psi)$ and the *poloidal current flux* $f = f(\psi)$. The poloidal magnetic flux $\psi := r\mathbf{A} \cdot \mathbf{e}_\varphi$ is the scaled toroidal component of the vector potential $\mathbf{A}$, i.e. $\mathbf{B} = \operatorname{curl} \mathbf{A}$ and $\mathbf{e}_\varphi$ the unit vector for φ. The poloidal current flux $f = r\mathbf{B} \cdot \mathbf{e}_\varphi$ is the scaled toroidal component of the magnetic field. We introduce $\mathbb{H} = [0, \infty] \times [-\infty, \infty]$, the positive half plane, to denote the meridian plane that contains the cross section of the tokamak device. The geometry of the tokamak determines the various subdomains (see Figure 1):

- the domain $\mathcal{F} \subset \mathbb{H}$ corresponds to those parts that are made of the ferromagnetic circuit,i.e. the iron core and return limbs;
- the domain $\mathcal{C}_i \subset \mathbb{H}$ corresponds to the M poloidal field coils, where each coil $\mathcal{C}_i$ has n_i wire turns, total resistance R_i and cross section $|\mathcal{C}_i|$, the coils are part of an electric circuit, that contains also N voltage suppliers;
- the domain $\mathcal{S} \subset \mathbb{H}$ corresponds to the passive structures, with conductivity σ;
- the domain $\mathcal{L} \subset \mathbb{H}$, bounded by the limiter, corresponds to the domain that is accessible by the plasma.

Then (see [3] or [12]), the equilibrium of plasma in a tokamak has to satisfy at each instant the following non-linear initial value problem:

$$\begin{aligned} L\psi(r, z, t) &= j(r, \psi(r, z, t), t) \quad \text{in} \quad \Omega\,; \\ \psi(0, z, t) &= 0\,; \\ \lim_{\|(r,z)\| \to +\infty} \psi(r, z, t) &= 0\,; \\ \psi(r, z, 0) &= \psi^0(r, z)\,, \end{aligned} \tag{5}$$

where L is a non-linear second-order elliptic differential operator

$$L\psi := -\frac{\partial}{\partial r}\left(\frac{1}{\mu[\psi]r}\frac{\partial \psi}{\partial r}\right) - \frac{\partial}{\partial z}\left(\frac{1}{\mu[\psi]r}\frac{\partial \psi}{\partial z}\right) := -\nabla \cdot \left(\frac{1}{\mu[\psi]r}\nabla \psi\right), \tag{6}$$

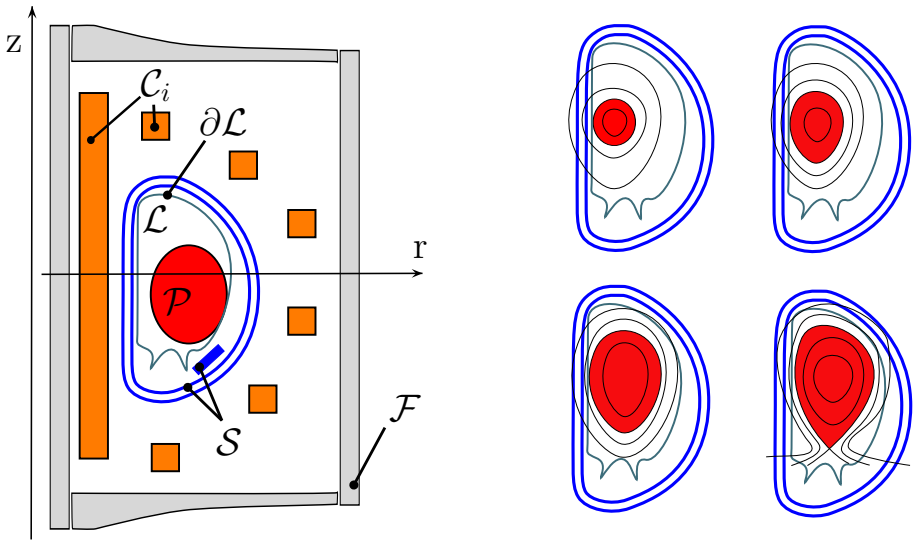


FIGURE 1. Left: Geometric description of the poloidal cross section of the tokamak device. Right: Sketch for characteristic plasma shapes during the so-called ramp-up phase. The ψ-isolines are indicated by black lines. In the beginning (first three pictures) the plasma touches the limiter and becomes more and more elongated (limiter configuration) while finally, it moves into the divertor configuration, where the plasma boundary contains an X-point of the poloidal flux.

with

$$\mu[\psi](r,z) = \mu_{\mathrm{f}}(|\nabla\psi(r,z)|^2 r^{-2}) \begin{cases} \geq \mu_0 & \text{in } \mathcal{F} \\ = \mu_0 & \text{elsewhere.} \end{cases} \tag{7}$$

Here, ∇ is the 2D gradient in the (r,z)-plane. The current density j is a non-linear function of ψ:

$$j(r,\psi(t),t) = \begin{cases} rp'(\psi(t),t) + \frac{1}{\mu_0 r} ff'(\psi(t),t) & \text{in } \mathcal{P}[\psi(t)]\,; \\ \frac{I_i}{|\mathcal{C}_i|} & \text{in } \mathcal{C}_i\,; \\ -\frac{\sigma}{r}\frac{\partial\psi(t)}{\partial t} & \text{in } \mathcal{S}\,; \\ 0 & \text{elsewhere}\,, \end{cases} \tag{8}$$

where the plasma section $\mathcal{P}[\psi(t)]$ is the domain bounded by the last closed poloidal magnetic flux line inside $\mathcal{L}$ and containing the magnetic axis $(r_{\mathrm{ax}}, z_{\mathrm{ax}})$. The domain $\mathcal{L}$ itself is bounded by the limiter and defines the domain that is accessible to the plasma. The magnetic axis is the point $(r_{\mathrm{ax}}, z_{\mathrm{ax}}) = (r_{\mathrm{ax}}[\psi], z_{\mathrm{ax}}[\psi])$, where ψ has its global maximum in $\mathcal{L}$. For convenience, we introduce also the coordinates $(r_{\mathrm{bd}}, z_{\mathrm{bd}}) = (r_{\mathrm{bd}}[\psi], z_{\mathrm{bd}}[\psi])$ of the point that determines the plasma boundary. $(r_{\mathrm{bd}}, z_{\mathrm{bd}})$ is either a hyperbolic point of ψ (in the case of a divertor configuration) or the contact point with the limiter $\partial\mathcal{L}$ (in the case of a limiter configuration). The different characteristic shapes of $\mathcal{P}(\psi)$ are illustrated in Figure 1 (right): the

boundary of $\mathcal{P}(\psi)$ either touches the boundary of $\mathcal{L}$, the limiter, or the boundary contains one or more hyperbolic points (X-points of ψ).

The current density j is non-linear in ψ due to the non-linear functions p', ff' as well as the definition of the plasma domain $\mathcal{P}(\psi)$.

The total coil current I_i in (8) relates to voltages in the suppliers and to self and mutual induction

$$\frac{I_i}{|\mathcal{C}_i|} = \sum_{j=1}^{n} \mathbf{R}_{ij} V_j(t) + \sum_{k=1}^{m} \mathbf{S}_{ik} \int_{\mathcal{C}_k} \frac{\partial \psi(t)}{\partial t}\, dr\, dz\,, \quad 1 \le i \le m\,, \tag{9}$$

via electric circuit equations. In the case of *simple electrical circuits*, e.g. circuits where each coil is connected to only one voltage supplier, we have

$$\mathbf{R}_{ij} = \begin{cases} 0 & i \neq j \\ \frac{n_i}{R_i |\mathcal{C}_i|} & i = j \end{cases} \quad \text{and} \quad \mathbf{S}_{ij} = \begin{cases} 0 & i \neq j \\ -\frac{2\pi n_i^2}{R_i |\mathcal{C}_i|^2} & i = j \end{cases}$$

and we refer to the Appendix B for the general case.

3. Numerical Methods

There are two different approaches to arrive at finite dimensional SQP formulations for the constrained optimization problem (1). In the first one, the *optimize-then-discretize* approach, one computes first the optimality conditions for the continuous optimization problem, and then discretizes them. The second approach, the *discretize-then-optimize* approach, discretizes directly the optimization problems. This yields a finite dimensional constrained optimization problem for which a SQP formulation follows immediately from the literature [15, 4, 11] (see also the Appendix A).

We prefer to work with the *discretize-then-optimize* approach for the following reasons: the *discretize-then-optimize* approach yields the exact gradient of the discrete objective function, while the *optimize-then-discretize* approach yields only an approximation. Both approaches involve approximation, but the *optimize-then-discretize* approach does not yield the exact gradient of either the continuous objective in (1), or the discretized objective [10]. Therefore, the *optimize-then-discretize* approach may result in inconsistent gradients, which, in turn, may cause serious convergence difficulties in the optimization process.

Hence, it remains to specify the discretization of non-linear constraints (5), (8) and (9) and the discretization of the cost functionals. Moreover, as we use SQP, we will also have to provide some details on first and second order derivatives of the discretized cost functionals and constraints. The implementation itself is kept flexible, so that cost functionals can be easily changed and modified. Adding new cost functionals encoding new design goals is simple.

For the discretization in time we introduce $N_T + 1$ not necessarily equidistant collocation points $T_0 \le t_s \le T_1$, $0 \le s \le N_T$. The discretization in space is based on the finite element method. Further details on the discretization of the constraints and the objective functional of (1) follow in Sections 3.1 and 3.2. The Section 3.3 bundles these two parts and presents the SQP formulation.

3.1. Free-Boundary Plasma Equilibrium Evolution.

The current density j in (8) is non-linear in ψ due to the non-linear functions p' and ff' and the definition of the plasma domain $\mathcal{P}(\psi)$. While $\mathcal{P}(\psi)$ is fully determined for a given ψ, the

two functions p' and ff' are not determined by the model (5)-(8). The model (5)-(8) needs to be augmented by the so-called transport and diffusion equations, that determine p' and ff'. In this work, we will assume that, up to some scaling coefficient λ, the functions p' and ff' are known. But, the domain of p' and ff' is the interval $[\psi_{\mathrm{bd}}, \psi_{\mathrm{ax}}]$ with the scalar values ψ_{ax} and ψ_{bd} being the flux values at the *magnetic axis* and at the boundary of the plasma:

$$\psi_{\mathrm{ax}}[\psi] := \psi(r_{\mathrm{ax}}[\psi], z_{\mathrm{ax}}[\psi])\,, \qquad \psi_{\mathrm{bd}}[\psi] := \psi(r_{\mathrm{bd}}[\psi], z_{\mathrm{bd}}[\psi])\,. \tag{10}$$

So, since the domain of p' and ff' depends on the poloidal magnetic flux itself, it is more practical to supply those profiles as functions of the normalized poloidal flux $\overline{\psi}(r,z)$:

$$\overline{\psi}(r,z) := \frac{\psi(r,z) - \psi_{\mathrm{ax}}[\psi]}{\psi_{\mathrm{bd}}[\psi] - \psi_{\mathrm{ax}}[\psi]}\,. \tag{11}$$

These two functions, subsequently termed $S_{p'}$ and $S_{ff'}$, have, independently of ψ, a fixed domain $[0,1]$. The functions $S_{p'}$ and $S_{ff'}$ are usually given as polynomial functions or piecewise polynomial functions. The scaling coefficients $\lambda(t)$ are determined by fixing the total plasma currents $\int_{\mathcal{P}(\psi)} j(r, \psi(r,z,t), t)\, drdz$ to some given values $I_p(t)$:

$$\lambda(t) \int_{\mathcal{P}(\psi)} r S_{p'}(\overline{\psi}(r,z,t), t) + \frac{1}{\mu_0 r} S_{ff'}(\overline{\psi}(r,z,t), t)\, drdz = I_p(t)\,.$$

Let $\Omega \subset \mathbb{H}$ be a sufficiently large semi-circle of radius ρ_Γ, that is centered at the origin and contains the geometry of the tokamak. The boundary $\partial\Omega$ splits into $\Gamma_{r=0} := \{(r,z)\,,\ r=0\}$ and $\Gamma = \partial\Omega \setminus \Gamma_{r=0}$. We introduce a triangulation Ω_h of the domain Ω that resolves the subdomains $\mathcal{L}, \mathcal{F}, \mathcal{C}_i, \mathcal{S}$ and use standard H^1-conforming Lagrangian finite elements with nodal degrees of freedom.

Let $b_k(r,z)$ denote the Lagrangian basis functions associated to the vertices of the mesh, e.g. b_k vanishes at all mesh vertices except one. Basis functions associated to vertices at $r=0$ are excluded from this finite element space X_h, as, due to axisymmetry $\psi(0,z) = 0$. The finite element approximation ψ_h of ψ at $t = t_s$ is:

$$\psi_h^s(r,z) = \sum_{k=1}^{|X_h|} \psi_{ks} b_k(r,z) \text{ with } \psi_{ks} \in \mathbb{R},\ 1 \le k \le |X_h|,\ 0 \le s \le N_T. \tag{12}$$

The *domain of the plasma* $\mathcal{P}(\psi_h)$ of a finite element function ψ_h is bounded by a continuous, piecewise straight, closed line. The critical points $(r_{\mathrm{ax}}(\psi_h), z_{\mathrm{ax}}(\psi_h))$ and $(r_{\mathrm{bd}}(\psi_h), z_{\mathrm{bd}}(\psi_h))$ are the coordinates of certain vertices of the mesh. The saddle point of a piecewise linear function ψ_h is some vertex (r_0, z_0) with the following property: if $(r_1,z_1), (r_2,z_2) \dots (r_N, z_N)$, denote the counterclockwise ordered neighboring vertices the sequence of discrete gradients $\psi_0 - \psi_1$, $\psi_0 - \psi_2 \dots \psi_0 - \psi_N$ changes at least four times the sign.

We get the following discretized version (modulo quadrature) of the free-boundary plasma equilibrium evolution: For given evolution $\vec{V}(t) = (V_1(t), \dots V_m(t))$ and $I_p(t)$ of the voltages and the total plasma current and for given initial conditions $\psi_h^0(r,z) = \sum_{k=1}^{|X_h|} \psi_{k0} b_k(r,z)$ we find for $1 \le s \le N_T$ the coefficients $\psi_{ks} \in \mathbb{R}, 1 \le$

$k \le |X_h|$ of ψ_h^s, the values $I_{is} \in \mathbb{R}, 1 \le i \le m$ and $\lambda^s \in \mathbb{R}$ such that:

$$
\begin{aligned}
&\int_\Omega \frac{1}{r\mu[\psi_h^s]} \nabla\psi_h^s \cdot \nabla\xi \, drdz - \lambda^s \int_{\mathcal{P}(\psi_h^s)} rS_{p'}(\overline{\psi}_h^s, t_s)\xi + \frac{S_{ff'}(\overline{\psi}_h^s, t_s)}{r\mu_0}\xi \, drdz \\
&- \sum_{i=1}^m \frac{I_{is}}{|\mathcal{C}_i|} \int_{\mathcal{C}_i} \xi drdz + \int_{\mathcal{S}} \frac{\sigma}{r} \frac{\psi_h^s - \psi_h^{s-1}}{t_s - t_{s-1}} \xi \, drdz + \mathsf{c}(\psi_h^s, \xi) = 0 \quad \forall \xi \in X_h \,, \\
&\sum_{j=1}^n \mathbf{R}_{ij} V_j(t_s) + \sum_{k=1}^m \mathbf{S}_{ik} \int_{\mathcal{C}_k} \frac{\psi_h^s - \psi_h^{s-1}}{t_s - t_{s-1}} \, drdz = \frac{I_{is}}{|\mathcal{C}_i|} \quad 1 \le i \le m \,, \\
&\lambda^s \int_{\mathcal{P}(\psi_h^s)} rS_{p'}(\overline{\psi}_h^s, t_s) + \frac{1}{r\mu_0} S_{ff'}(\overline{\psi}_h^s, t_s) \, drdz = I_P^s \,.
\end{aligned} \tag{13}
$$

The bilinear form $\mathsf{c} : V \times V$ on Γ, accounts for the boundary conditions at infinity [1]:

$$
\begin{aligned}
\mathsf{c}(\psi, \xi) :=& \frac{1}{\mu_0} \int_\Gamma \psi(\mathbf{P}_1) N(\mathbf{P}_1) \xi(\mathbf{P}_1) dS_1 \\
&+ \frac{1}{2\mu_0} \int_\Gamma \int_\Gamma (\psi(\mathbf{P}_1) - \psi(\mathbf{P}_2)) M(\mathbf{P}_1, \mathbf{P}_2)(\xi(\mathbf{P}_1) - \xi(\mathbf{P}_2)) dS_1 dS_2.
\end{aligned} \tag{14}
$$

with

$$
\begin{aligned}
M(\mathbf{P}_1, \mathbf{P}_2) &= \frac{k_{\mathbf{P}_1,\mathbf{P}_2}}{2\pi(r_1 r_2)^{\frac{3}{2}}} \left(\frac{2 - k^2_{\mathbf{P}_1,\mathbf{P}_2}}{2 - 2k^2_{\mathbf{P}_1,\mathbf{P}_2}} E(k_{\mathbf{P}_1,\mathbf{P}_2}) - K(k_{\mathbf{P}_1,\mathbf{P}_2}) \right) \\
N(\mathbf{P}_1) &= \frac{1}{r_1} \left(\frac{1}{\delta_+} + \frac{1}{\delta_-} - \frac{1}{\rho_\Gamma} \right) \text{ and } \delta_\pm = \sqrt{r_1^2 + (z_1 - \rho_\Gamma)^2} \,,
\end{aligned}
$$

where $\mathbf{P}_i = (r_i, z_i)$ and K and E are the complete elliptic integrals of first and second kind, respectively and

$$
k_{\mathbf{P}_j,\mathbf{P}_k} = \sqrt{\frac{4 r_j r_k}{(r_j + r_k)^2 + (z_j - z_k)^2}} \,.
$$

We refer to [9, Chapter 2.4] for the details of the derivation. Alternative approaches, that incorporate the boundary conditions at infinity were recently presented in [7].

Quadrature rules. The integration over the domain Ω is split into a sum of integrals over the triangles T of the mesh and we use the barycenter quadrature rule to approximate these integrals: For a triangle with vertex coordinates $\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k \in \mathbb{R}^2$ the quadrature point is the barycenter $\mathbf{b}_T := \frac{1}{3}(\mathbf{a}_i + \mathbf{a}_j + \mathbf{a}_k)$ and the quadrature weight ω_T is the size $|T|$ of the triangle.

Besides the integrals over Ω, the weak formulation (13) involves also integrals over the plasma domain $\mathcal{P}(\psi_h)$. As the mesh does not resolve the boundary of the plasma domain $\mathcal{P}(\psi_h)$, we need to specify also the quadrature rule that is used to approximate integrals over intersections $T \cap \mathcal{P}(\psi_h)$ of triangles with the plasma domain. We use again barycenter quadrature, but here the quadrature point and weight will depend non-linearly on ψ_h, which needs to be taken into account when we use linearizations of (13), e.g. in Newton or SQP iterations. The technical details of such linearizations can be found in [10].

The line integrals over Γ in the definition of $\mathsf{c}(\cdot, \cdot)$ are split into line integrals over edges on triangles whose vertices are on Γ. Then, the trapezoidal quadrature rule yields an approximation of sufficient accuracy.

3.2. **Cost- and Regularization Functionals.** As we already specified the approximation of ψ there is not much left to explain on the discretization of cost functionals that involve the poloidal flux and the voltages.

Here, the same barycentric quadrature rule as in 3.1 is used to approximate area integrals, while time integrals are approximated by the composite trapezoidal rule. This yields expressions that are algebraic in the principal unknowns $\psi_{ks}, 1 \leq k \leq |X_h|, 1 \leq s \leq N_T$ and $I_{is}, 1 \leq i \leq m, 1 \leq s \leq N_T$. So, computation and implementation of first and second order derivatives of such functionals become a mechanical iterated application of the usual rules of differential calculus.

The discretization of the regularization functionals likewise uses composite trapezoidal rule for time integrals. Moreover, we represent each voltage as a finite series

$$V_i(t) = \sum_{j=1}^{N_c} V_{ij} B_j(t), \quad 1 \leq i \leq n\,,$$

where the $B_j(t)$ are the basis functions of a polynomial or spline space over $[0, T]$. So again this yields expressions that are algebraic in the unknowns V_{ij}, $1 \leq i \leq n$, $1 \leq j \leq N_c$.

3.3. **SQP Formulation.** Combining the discretized free-boundary plasma equilibrium evolution (13) with discretized objective and regularization functionals we arrive at discretized versions of the optimal control formulation (1) that is of the general form

$$\min_{\mathbf{u},\mathbf{y}} J(\mathbf{y}, \mathbf{u}) \quad \text{s.t.} \quad \mathbf{b}(\mathbf{y}, \mathbf{u}) = 0\,. \tag{15}$$

The state variable $\mathbf{y}$ contains the unknowns ψ_{ks} $1 \leq k \leq |X_h|$, $1 \leq s \leq N_T$ of the poloidal flux and the scaling parameter λ^s, $1 \leq s \leq N_T$, hence $\mathbf{y} \in \mathbb{R}^{(|X_h|+1)N_T}$. Likewise the constraint $\mathbf{b}(\mathbf{y}, \mathbf{u}) = 0$ corresponds to $(|X_h|+1)N_T$ coupled non-linear equations as the unknowns for the currents I_{is}, $1 \leq i \leq m$, $1 \leq s \leq N_T$ have been eliminated from the formulation (13). The control variable $\mathbf{u}$ contains a subset of the expansion coefficients V_{ij}, $1 \leq i \leq n$, $1 \leq j \leq N_c$ for the voltages. One could for example prescribe the voltage of a few suppliers and treat only the voltages of the remaining suppliers as unknowns. Another possibility would be to treat only certain coefficients as unknown. E.g. if one works with an expansion in a hierarchical basis, it would be beneficial to prescribe the coefficients for low order terms and keep only coefficients of higher order polynomials. So, in general $\mathbf{u} \in \mathbb{R}^N$ with $1 \leq N \leq N_c n$.

The *SQP* formulation for the constrained optimization problem (15) involves first and second order derivatives of $\mathbf{b}(\mathbf{y}, \mathbf{u})$ and $J(\mathbf{y}, \mathbf{u})$. But, as we presented in Sections 3.1 and 3.2 explicit expressions for $\mathbf{b}(\mathbf{y}, \mathbf{u})$ and $J(\mathbf{y}, \mathbf{u})$ that are algebraic in $\mathbf{u}$ and $\mathbf{y}$, we can also provide explicit expressions for the first and second order derivatives. An inspection of (13) shows that the constraint $\mathbf{b}(\mathbf{y}, \mathbf{u})$ is affine in the control unknown $\mathbf{u}$, hence has vanishing second order derivatives. Moreover, to avoid the expensive assembling of second order derivatives of $\mathbf{b}(\mathbf{y}, \mathbf{u})$ we neglect those in the SQP iterations. In the terminology of Newton methods we use rather a *quasi* SQP method, than an *exact* SQP method. It is known that such modifications are prone to convergence issues [4], but this doesn't seem to be an issue for our specific application. The Algorithm 1 summarizes the quasi SQP formulation, and should be compared to the exact SQP in Algorithm 4 (see Appendix A). We want to

stress that the size of $\mathbf{Y}$ in Algorithm 4 scales at least linearly (and quadratically if $N_c = O(N_T)$) with the number of time steps N_T, which causes memory to be the limiting factor for computations with a very large number of time steps. Nevertheless, with the current implementation we can go easily beyond 200 and more time steps, which is sufficient for the applications in mind. Moreover, we avoid the introduction of additional parameters due to iterative solver and make use of fast methods for linear systems with multiple righthand sides.

Algorithm 1 SQP (quasi) with direct solver

$\Delta\mathbf{u} \leftarrow 1$, $\Delta\mathbf{y} \leftarrow 1$, $\mathbf{y} \leftarrow \mathbf{y}^0$, $\mathbf{u} \leftarrow \mathbf{u}^0$
while $\|\Delta\mathbf{u}\|/\|\mathbf{u}\| > tol$, $\|\Delta\mathbf{y}\|/\|\mathbf{y}\| > tol$ **do**
  $(\Delta\mathbf{y}, \mathbf{Y}) \leftarrow -\mathbf{b}_\mathbf{y}^{-1}(\mathbf{y},\mathbf{u})(\mathbf{b}(\mathbf{y},\mathbf{u}), \mathbf{b}_\mathbf{u}(\mathbf{y},\mathbf{u}))$
  $\mathbf{m}_{01} \leftarrow J_\mathbf{u}^T(\mathbf{y},\mathbf{u})$, $\mathbf{m}_{10} \leftarrow \mathbf{Y}^T J_\mathbf{y}^T(\mathbf{y},\mathbf{u})$
  $\mathbf{M}_{02} \leftarrow J_{\mathbf{uu}}(\mathbf{y},\mathbf{u})$
  $\mathbf{m}_{11} \leftarrow J_{\mathbf{uy}}(\mathbf{y},\mathbf{u})\Delta\mathbf{y}$, $\mathbf{M}_{11} \leftarrow J_{\mathbf{uy}}(\mathbf{y},\mathbf{u})\mathbf{Y}$
  $\mathbf{m}_{20} \leftarrow \mathbf{Y}^T J_{\mathbf{yy}}(\mathbf{y},\mathbf{u})\Delta\mathbf{y}$, $\mathbf{M}_{20} \leftarrow \mathbf{Y}^T J_{\mathbf{yy}}(\mathbf{y},\mathbf{u})\mathbf{Y}$
  $\mathbf{m} \leftarrow \mathbf{m}_{01} + \mathbf{m}_{10} + \mathbf{m}_{11} + \mathbf{m}_{20}$
  $\mathbf{M} \leftarrow \mathbf{M}_{02} + \mathbf{M}_{11} + \mathbf{M}_{11}^T + \mathbf{M}_{20}$
  $\Delta\mathbf{u} \leftarrow -\mathbf{M}^{-1}\mathbf{m}$
  $\mathbf{y} \leftarrow \mathbf{y} + \Delta\mathbf{y} + \mathbf{Y}\Delta\mathbf{u}$
  $\mathbf{u} \leftarrow \mathbf{u} + \Delta\mathbf{u}$
end while

4. Numerical Validation Tests

We present different numerical tests, that verify that the implementation of derivatives of the non-linear constraints (13) in FEEQS.M is correct. These derivatives are the cornerstone of the proposed optimal control approach and appear through $\mathbf{b}_\mathbf{y}(\mathbf{y},\mathbf{u})$ in SQP (see Algorithm 1).

The following calculations are based on a ITER-like geometry and an equlibrium plasma (see Fig. 2) that corresponds to the currents in the table of Figure 2. The total plasma current is $I_\mathrm{P} = 15 \times 10^6 A$ and the two profiles $S_{p'}$ and $S_{ff'}$ in the definition of the plasma current density are

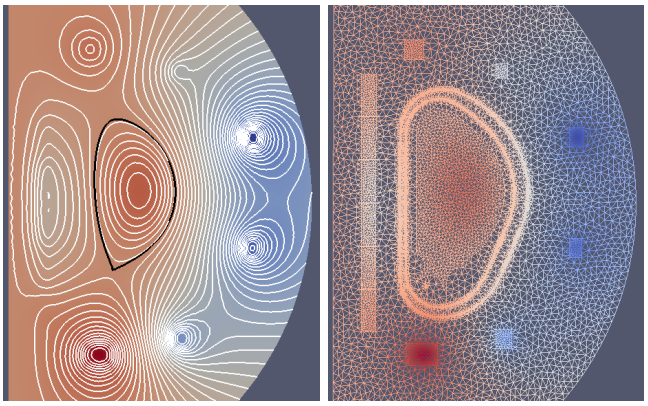
$$S_{p'}(y) = \frac{\beta}{r_0}((1-y)^\alpha)^\gamma\,, \quad S_{ff'}(y) \quad = (1-\beta)\mu_0 r_0 (1-y^\alpha)^\gamma\,, \tag{16}$$

with the four parameters $r_0 = 6.2m$, $\alpha = 2.0$, $\beta = 0.5978$ and $\gamma = 1.395$.

4.1. **Derivatives due to the free-boundary.** Let $D\mathsf{J}_h(\psi_h) \in \mathbb{R}^{|X_h|\times|X_h|}$ denote the derivative of the discretization $\mathsf{J}_h(\psi_h, b_m)$ (via quadrature from Section 3.1) of the non-linear mapping

$$\mathsf{J}(\psi_h, b_m) = \int_{\mathcal{P}(\psi_h)} \left[r S_{p'}(\overline{\psi}_h) + \frac{1}{r\mu_0} S_{ff'}(\overline{\psi}_h, t_s) \right] b_m \, dr dz \quad 1 \le m \le |X_h|\,. \tag{17}$$

This mapping involves the non-linearity due to the free plasma boundary. The linearization is non-standard [10]. In this first test it is verified that for given



coil	$I_i[At]$
U3	-1.400×10^6
U2	-9.500×10^6
U1	-20.388×10^6
L1	-20.388×10^6
L2	-9.000×10^6
L3	3.564×10^6
P1	5.469×10^6
P2	-2.266×10^6
P3	-6.426×10^6
P4	-4.820×10^6
P5	-7.504×10^6
P6	17.240×10^6

FIGURE 2. Left: The plasma (flux lines and flux intensity) that correspond to the currents in the table. Center: The triangulation of the ITER-geometry. Right: Data for coils. Coils U1-U3 and L1-L3 are upper and lower central solinoid coils. Coils P1-P6 are the poloidal field coils.

$\psi_h(r,z) = \sum_{k=1}^{|X_h|} \psi_k b_k(r,z)$ and perturbation $\delta_h(r,z) = \sum_{k=1}^{|X_h|} \delta_k b_k(r,z)$ the implementation of $D\mathsf{J}_h(\psi_h)$ yields:

$$E_{\mathrm{FD}}(\varepsilon) := \frac{\left\| \varepsilon^{-1} \left(\mathsf{J}_h(\psi_h + \varepsilon\delta_h, \cdot) - \mathsf{J}_h(\psi_h, \cdot) \right) - D\mathsf{J}_h(\psi_h) \cdot \delta \right\|}{\| D\mathsf{J}_h(\psi_h) \cdot \delta \|} = O(\varepsilon). \tag{18}$$

The perturbation increment δ is randomly chosen. In Table 1 we monitor this relative error and observe, as expected, first order convergence.

4.2. **Convergence of the Sensitivities.** Another way to validate the implementation of the derivatives appearing in $\mathbf{b}_{\mathbf{y}}(\mathbf{y},\mathbf{u})$ would consist in solving the nonlinear constraint $\mathbf{b}(\mathbf{y},\mathbf{u})$ for $\mathbf{y}$ and monitor the convergence. For an accurate implementation of the derivatives the convergence will be quadratic. On the other hand, since in many cases also with inaccurate derivatives one will observe super-linear convergence, this test could fail as indicator for accurate derivatives. The following test is more appropriate: We consider a control $\mathbf{u}_\varepsilon$ that is parametrized by $\varepsilon \in \mathbb{R}$ and suppose we know $\mathbf{y}_0$ that verifies $\mathbf{b}(\mathbf{y}_0, \mathbf{u}_0)$. Then it holds

$$\mathbf{y}_\varepsilon^1 - \mathbf{y}_\varepsilon = O(\varepsilon^2)\,, \tag{19}$$

where $\mathbf{y}_\varepsilon^1$ is the first Newton iteration for the problem $\mathbf{b}(\mathbf{y}_\varepsilon, \mathbf{u}_\varepsilon) = 0$ with initial guess $\mathbf{y}_0$:

$$\mathbf{b}_{\mathbf{y}}(\mathbf{y}_0, \mathbf{u}_\varepsilon) \cdot (\mathbf{y}_\varepsilon^1 - \mathbf{y}_0) = -\mathbf{b}(\mathbf{y}_0, \mathbf{u}_\varepsilon) \tag{20}$$

The result (19) follows from $\mathbf{y}_\varepsilon = \mathbf{y}_0 + \mathbf{y}_0'\varepsilon + O(\varepsilon^2)$ and $0 = \mathbf{b}(\mathbf{y}_0, \mathbf{u}_\varepsilon) + \varepsilon \mathbf{b}_{\mathbf{y}}(\mathbf{y}_0, \mathbf{u}_\varepsilon)\mathbf{y}_0' + O(\varepsilon^2)$. When the exact derivative $\mathbf{b}_{\mathbf{y}}(\mathbf{y}_0, \mathbf{u}_\varepsilon)$ in (20) is replaced by some approximation $\overline{\mathbf{A}}$:

$$\overline{\mathbf{A}} \cdot (\overline{\mathbf{y}}_\varepsilon^1 - \mathbf{y}_0) = \mathbf{f}(\mathbf{u}_\varepsilon) - \mathbf{a}(\mathbf{y}_0)\,.$$

then $\overline{\mathbf{y}}_\varepsilon^1$ will be at most first order approximation of $\mathbf{y}_\varepsilon$.

The first test of this kind is for the static free-boundary plasma equilibrium (FBE) problem with the following Galerkin formulation: For given currents $I_1, \ldots I_M$:

Find ψ_h such that

$$\begin{aligned}\int_\Omega \frac{1}{r\mu[\psi_h]}\nabla\psi_h\cdot\nabla\xi\,drdz - \lambda\int_{\mathcal{P}(\psi_h)} rS_{p'}(\overline{\psi}_h)\xi + \frac{S_{ff'}(\overline{\psi}_h)\xi}{r\mu_0}\,drdz \\ = \sum_{i=1}^{M}\frac{I_i}{|\mathcal{C}_i|}\int_{\mathcal{C}_i}\xi drdz \quad \forall\xi\in X_h \\ \lambda\int_{\mathcal{P}(\psi_h)} rS_{p'}(\overline{\psi}_h) + \frac{S_{ff'}(\overline{\psi}_h)}{r\mu_0}\,drdz = I_P\,.\end{aligned} \tag{21}$$

The discretization and linearization follow as described in Section 3.1 and 3.3. Then the unperturbed control data $\mathbf{u}_0$ is the data given in the table in Figure 2 and perturbation is a random incremental current for each coil scaled with $\varepsilon = 0.5^0,\dots 0.5^{14}$. In Table 1 (4th column) the error $E_{\mathrm{DN}}(\varepsilon) = \|\mathbf{y}_\varepsilon - \mathbf{y}_\varepsilon^1\|$ is monitored. We observe second order convergence, which shows that we use accurate derivatives. In contrast, the result $E_{\mathrm{CN}}(\varepsilon) = \|\mathbf{y}_\varepsilon - \overline{\mathbf{y}}_\varepsilon^1\|$ for a Newton-type iteration method (see 6th column in Table 1) that follows from the discretization of derivatives of the continuous problem [3, 9] yields only first order convergence.

Repeating the same test for the implementation for FBEE problem (13), based on the discretization and linarization described in Section 3.1 and 3.3 we observe the expected second order convergence (see Table 1, 8th and 9th column). This reassures that the implementation in FEEQS.M is correct.

5. Applications for tokamaks WEST and HL-2M

5.1. **Fast limiter to divertor transition for the WEST tokamak.** In this section we give tests with different objective functions and regularization terms for the WEST tokamak [5], which is a new tokamak in CEA Cadarache that aims at testing the ITER-like technology for tungsten (W) plasma facing components. Exploring scenarios which can handle fast transitions from limiter to divertor configuration within the engineering limits, such as the maximum and minimum voltages (in Table 2) that can be handled by the coils power supply systems may be a way of reducing plasma pollution by W and will be our goal in the following simulations.

The Figure 3 shows the geometrical setting of WEST. All the results are based on a mesh with 52210 triangles and 27763 vertices. We use $N_T = 10$ time steps with equidistant time step length of $5\,ms$, the voltage in A coil is fixed to its maximum value of $1400\,V$ in all the following cases to provide energy for flux consumption, and the degree of the polynomial representation of the voltage is 8. The evolution of the total plasma current $I_p(t)$ and the parameters $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ for the current density profiles $S_{p'}$ and $S_{ff'}$ in (16) are based on experimental data of Tore Supra, the predecessor of WEST.

5.1.1. *Prescribed level sets at all time steps.* As a first approach to the problem, we solve the discretized inverse evolution problem (15) with $J(\mathbf{y},\mathbf{u}) := C(\mathbf{y}) + R_1(\mathbf{u})$,

$\varepsilon_i = \frac{1}{2}^i$	J in (17)		FBE in (21)				FBEE in (13)	
i	$E_{\mathrm{FD}}(\varepsilon_i)$	rate	$E_{\mathrm{DN}}(\varepsilon_i)$	rate	$E_{\mathrm{CN}}(\varepsilon_i)$	rate	$E_{\mathrm{DN}}(\varepsilon_i)$	rate
0	0.0198034		11647.2		2928.07		645.401	
1	0.0113253	0.81	5485.42	1.09	1125.83	1.38	450.549	0.52
2	0.0061465	0.88	3362.31	0.70	1182.56	−0.07	31.6012	3.83
3	0.0032184	0.93	1936.28	0.80	846.373	0.50	36.9223	−0.22
4	0.0016494	0.96	694.294	1.48	149.343	2.50	10.7593	1.78
5	0.0008353	0.98	4.26942	7.35	268.207	−0.84	2.66903	2.01
6	0.0004204	0.99	9.65561	−1.18	145.894	0.88	0.66434	2.01
7	0.0002109	1.00	2.39544	2.01	70.5145	1.05	0.16571	2.00
8	0.0001056	1.00	0.60127	1.99	34.6608	1.02	0.04138	2.00
9	0.0000528	1.00	0.15064	2.00	17.1804	1.01	0.01034	2.00
10	0.0000264	1.00	0.03770	2.00	8.55259	1.01	0.00258	2.00
11	0.0000132	1.00	0.00943	2.00	4.26687	1.00	0.00064	2.00
12	0.0000066	1.00	0.00236	2.00	2.13108	1.00	0.00016	2.00
13	0.0000033	1.00	0.00059	2.00	1.06495	1.00	0.00004	2.00
14	0.0000016	1.00	0.00015	2.00	0.53233	1.00	0.00001	2.00

TABLE 1. Convergence of the errors $E_{\dots}$ and the convergence rate $(\frac{log(E_{\dots}(\varepsilon_{i+1}))-log(E_{\dots}(\varepsilon_i))}{log(\varepsilon_{i+1})-log(\varepsilon_i)})$: 1.) the finite difference error E_{FD} (18); 2.) the error E_{DN} using the derivatives outlined in Sections 3.1 and 3.3 for the FBE problem (21); 3.) the error E_{CN} using derivatives in [3, 9] for the FBE problem (21); 4.) the error E_{DN} using the derivatives outlined in Sections 3.1 and 3.3 for the FBEE problem (13).

Coil	Maximum Voltage (V)	Minimum Voltage (V)
A	1400	−1400
Bh	1400	−1400
Dh	2500	−2500
Eh	2500	−2500
Fh	2500	−2500
Fb	2500	−2500
Eb	2500	−2500
Db	2500	−2500
Bb	1400	−1400
X_U	300	−300
X_L	300	−300

TABLE 2. Maximum and minimum voltages on the coils systems of WEST.

in which:

$$
\begin{aligned}
C(\mathbf{y}) &:= \frac{1}{2}\sum_{s=1}^{N_T}\sum_{i=1}^{N_{\mathrm{desi}}} w_s\big(\psi_h^s(r_i,z_i)-\psi_h^s(r_0,z_0)\big)^2\,,\\
R_1(\mathbf{u}) &:= \frac{1}{2}w_{R_1}\sum_{i=1}^{L}\sum_{s=1}^{N_T} V_i(t_s)^2\,.
\end{aligned}
\tag{22}
$$

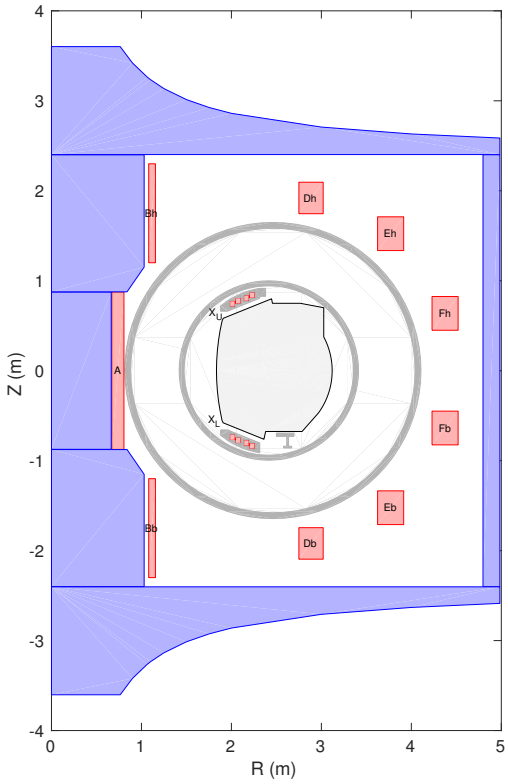


FIGURE 3. Cross-section of the WEST tokamak, red rectangles are the poloidal coils $\mathcal{C}_i$ and iron areas $\mathcal{F}$ are in purple, the dark grey areas are the vacuum vessels, the divertor (X) coils casings and the passive plates $\mathcal{S}$.

Here, $C(\mathbf{y})$ is the discretization of the cost function (2) introduced in Section 1, and has the purpose to prescribe a desired level set for every time step, and the weights are $w_i = 1, 1 \leq i \leq N_T$. The weight of the regularization term $R_1(\mathbf{u})$ is fixed as $w_{R_1} = 1 \times 10^{-11}$ during this calculation.

The results (see Figure 4) show that the prescribed level sets coincide fairly well with level sets of the numerical solution at all times, with a tiny discrepancy when the plasma evolves from limiter to divertor configuration. The evolution of the voltages can be seen in Figure 5, which shows that the voltage limits (the dashed lines) at some time steps are violated.

5.1.2. *Prescribed level sets at all times and constraint penalization term.* Secondly, we add two regularization terms to the objective function to penalize the violation of the voltage limits. E.g. we solve now the constrained optimization problem (15)

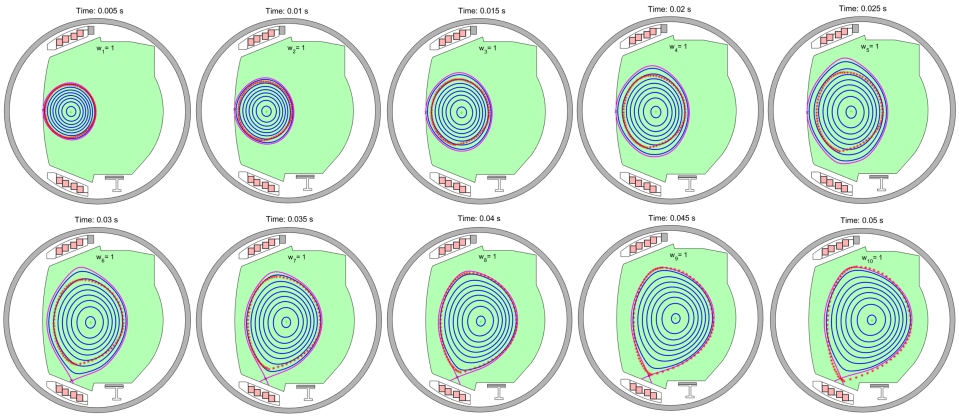


FIGURE 4. Prescribed level sets at all time steps (see 5.1.1): the magenta contour is the plasma boundary and the blue lines are the ψ level sets inside the plasma, the red points are the desired boundaries.

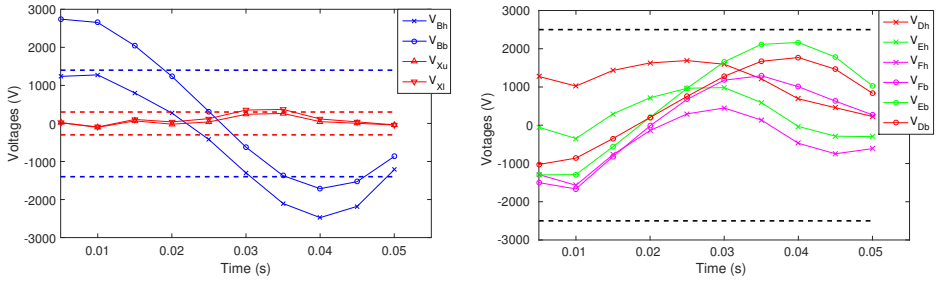


FIGURE 5. Prescribed level sets at all time steps (see 5.1.1): The voltage evolution, dashed lines are the limits.

for $J(\mathbf{y},\mathbf{u}) := C(\mathbf{y}) + R_1(\mathbf{u}) + R_2(\mathbf{u}) + R_3(\mathbf{u})$ with

$$
\begin{aligned}
R_2(\mathbf{u}) &:= \frac{1}{2}\sum_{i=1}^{L} w_{R_{2,i}} \sum_{s=1}^{N_T} max(V_i(t_s) - V_{i,max}, 0)^3\,, \\
R_3(\mathbf{u}) &:= \frac{1}{2}\sum_{i=1}^{L} w_{R_{3,i}} \sum_{s=1}^{N_T} max(V_{i,min} - V_i(t_s), 0)^3\,,
\end{aligned}
\tag{23}
$$

where the weights are $w_{R_{2,i}} = w_{R_{3,i}} = 1 \times 10^{-16}$.

The level sets of the flux inside and near the plasma are found to be almost the same as in the previous case (shown in Figure 4). But the evolution of the voltages, shown in Figure 6, is slightly different and almost all voltages remain within their limits.

5.1.3. *Prescribed level sets at start and end.* Assuming that level sets are prescribed at every time step is not very relevant in practical applications, where one aims to solve problems that require a few hundred or thousands of time steps. Providing level sets a each time step is a humongous amount of work that moreover confines

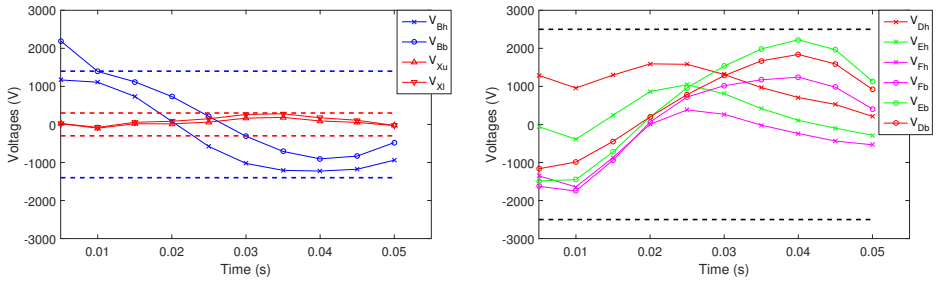


FIGURE 6. Prescribed level sets at all time steps and constraint penalization term (see 5.1.2): The voltage evolution, dashed lines are the limits.

the plasma evolution fairly strictly and could prevent from finding the best scenarios. The more relevant task asks to find reasonable trajectories of the flux for prescribed level sets at the beginning and end only. Therefore, we use in this third case the prescribed boundaries only on some of the time steps, meaning we set in the objective function $C(\mathbf{y})$ in (22) the weights w_1, w_2, w_3 and w_{10} to one but the remaining weights $w_4, w_5, ...w_9$ to zero. Introducing

$$C_0(\mathbf{y}) := \frac{1}{2} \sum_{s=\{1,2,3,10\}} \sum_{i=1}^{N_{\text{desi}}} \left(\psi_h^s(r_i, z_i) - \psi_h^s(r_0, z_0)\right)^2$$

for brevity, we solve hence here the optimization problem (15) with $J(\mathbf{y}, \mathbf{u}) := C_0(\mathbf{y}) + R_1(\mathbf{u}) + R_2(\mathbf{u}) + R_3(\mathbf{u})$, where the weights in the regularization terms $R_1(\mathbf{u})$, $R_2(\mathbf{u})$ and $R_3(\mathbf{u})$ are as in the previous case.

The plasma trajectory of the numerical solution is shown in Figure 7. It is found that the boundaries of the plasma match with the prescribed level sets in the beginning and end. But in between, when there is no prescribed level set, the plasma position and shape is changing drastically. The evolution of the voltages are presented in Figure 8, which shows that the voltages in the time steps without prescribed boundaries are within the bounds and the voltages in the divertor coils are violating the limits only during the last time step.

We obtained a solution that is reasonable for this particular formulation of the optimal control problem for scenario design: As the formulation does not contain any objective for the flux in between the start and the end, we risk in computing solutions with fast changes of plasma position and shape. On the other hand, scenarios with such fast changes are highly undesirable as they are very difficult to control in the experiment. The formulation in the next section is more adequate.

5.1.4. *Prescribed level sets at start and end with penalization term on induced currents.* Fast changes of the shape and position of the plasma lead to large induced currents in passive structures $\mathcal{S}$. In order to avoid the fast changes obtained in the previous simulations we add to the objective a new term that penalizes the induced currents

$$I_{ps,s}(\mathbf{y}) := \int_{\mathcal{S}} \frac{\sigma}{r} \frac{\psi_h^{s+1}(r,z) - \psi_h^s(r,z)}{t_{s+1} - t_s} dr\, dz \quad 1 \le s \le N_T$$

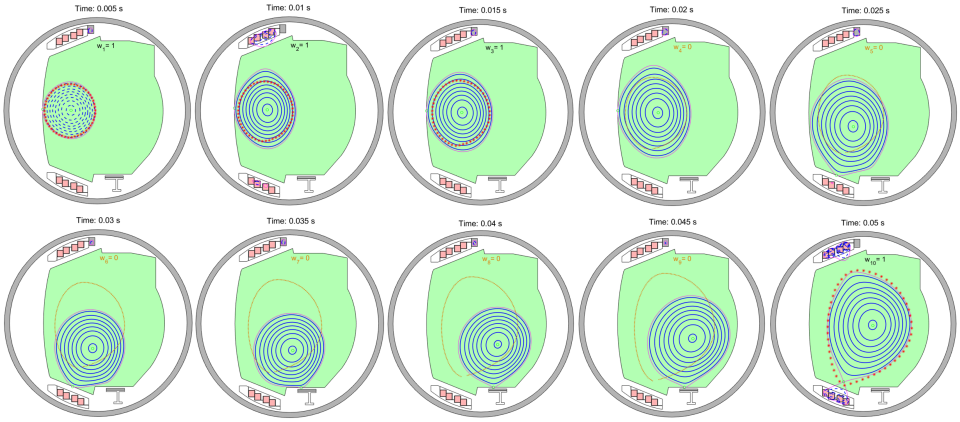


FIGURE 7. Prescribed level sets at start and end (see 5.1.3): the magenta contour is the plasma boundary and the blue lines are the ψ level set inside plasma, the red points are the desired boundaries.

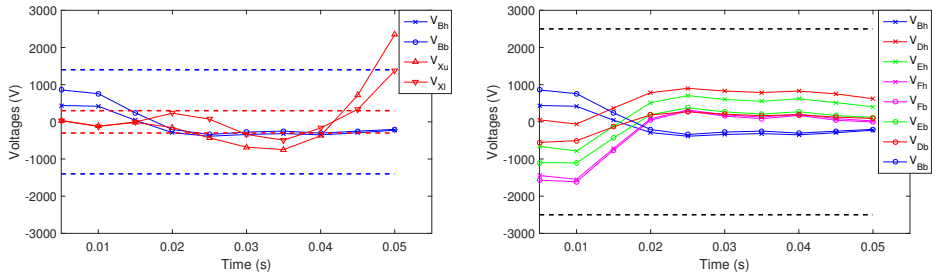


FIGURE 8. Prescribed level sets at start and end (see 5.1.3): The voltage evolution, dashed lines are the limits.

in passive structures $\mathcal{S}$ with conductivity σ. We solve the optimization problem (15) with $J(\mathbf{y},\mathbf{u}) := C_0(\mathbf{y}) + C_1(\mathbf{y}) + R_1(\mathbf{u}) + R_2(\mathbf{u}) + R_3(\mathbf{u})$ and

$$C_1(\mathbf{y}) = \frac{1}{2} w_{C_1} \sum_{s=1}^{N_T} I_{ps,s}^2(\mathbf{y}) \tag{24}$$

where $w_{C_1} = 1 \times 10^{-6}$.

The results, shown in Fig. 9 are quite satisfying: the plasma stays near to the equatorial plane even during the time steps when the level set is not prescribed and evolves in a smooth fashion to its final shape. This shows that the penalization term on induced currents allows finding slowly moving solutions even if we prescribed level sets only in the beginning and end. The evolution of the voltages is shown in Figure 10 and we highlight that all the voltages remain within the limits.

These tests show that with different objective functions and regularization terms, we can get a reliable plasma limiter to divertor transition while respecting the engineering limits (only maximum and minimum voltages where considered in this case).

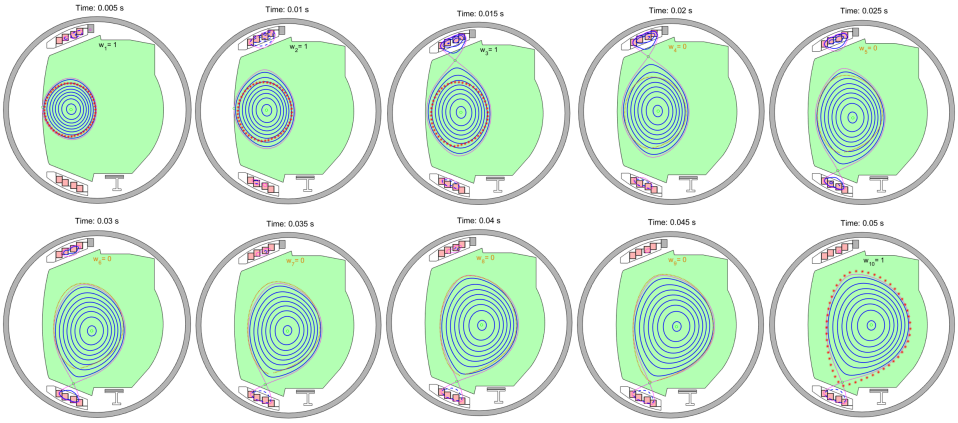


FIGURE 9. Prescribed level sets at start and end with penalization term on induced currents (see 5.1.4): the magenta contour is the plasma boundary and the blue lines are the ψ level sets inside the plasma, the red points are the desired boundaries.

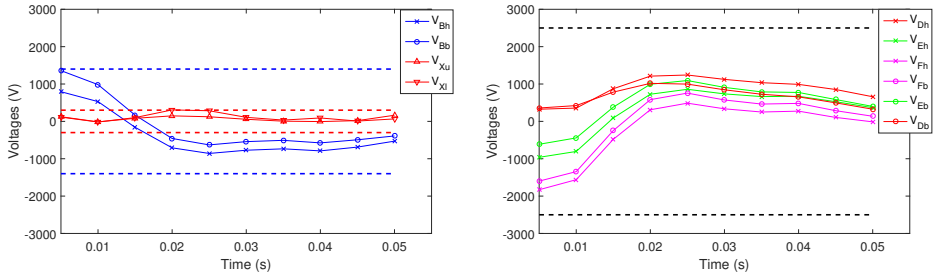


FIGURE 10. Prescribed level sets at start and end with penalization term on induced currents (see 5.1.4): The voltage evolution, dashed lines are the limits.

5.2. **Full scenario computation for HL-2M tokamak.** We give in this section a calculation for a full scenario, which includes the plasma current ramp up, flat top and ramp down phases, for the new tokamak of HL-2M [13] (see Figure 11 for the cross-section) in China. A major objective of this machine is the exploration of ITER related physical issues.

The settings for the computations are the following: The parameters I_p, α, β and γ are prescribed as in Figure 12. The length of time steps in the ramp up phase (from $0.1s$ to $2s$) and ramp down phase(from $8.1s$ to $10s$) is $0.1s$, while in the flat top phase (from $2.3s$ to $8s$) it is $0.3s$. In total we have 60 time steps. The objective function is $J(\mathbf{y}, \mathbf{u}) = C(\mathbf{y}) + R(\mathbf{u})$, with $C(\mathbf{y})$ the discretization of (2) and $R(\mathbf{u})$ the discretization of (3). Plasma boundaries are prescribed for every time step and the weights w and $\mathbf{D_{ii}}$ are 1 and 1×10^{-10} and zero otherwise. The computational mesh is divided into 30583 triangles, and the total number of vertices is 15367, and the degree of polynomial to fit the voltage is again fixed as 8.

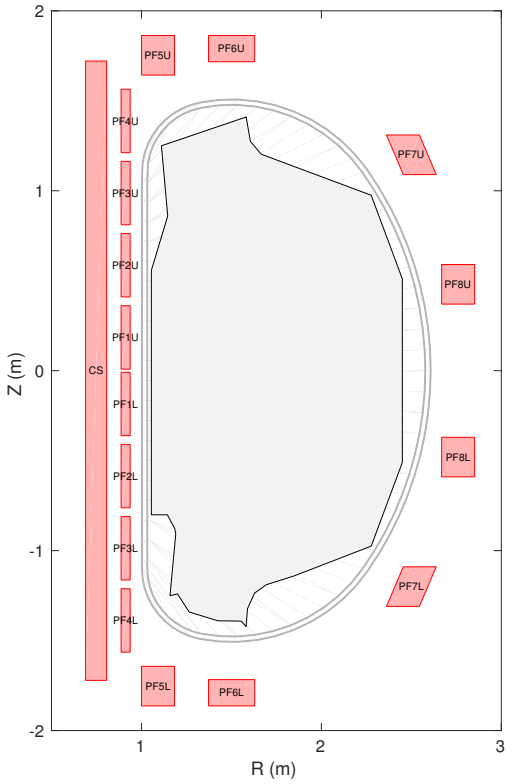


FIGURE 11. Cross-section of the HL-2M tokamak, red rectangles are the poloidal coils $\mathcal{C}_i$, the grey areas are the two layers vacuum vessel, and the black line is the limiter $\mathcal{L}$.

Altogether we end up with a constrained optimization problem that has almost one million unknowns of which approximately a hundred correspond to control unknowns $\mathbf{u}$. We solve this problem in 15 iterations and the computing time is less than $350\,s$ on standard notebook with a $4 \times 2.7\,GHz$ processor and $32\,GB$ memory.

In Figure 13 we show plasma boundaries at some selected time steps. As can be seen, the plasma follows well with reference boundary in the whole time interval. The voltages are shown in Figure 14. This example is a showcase that the presented approach can construct a full operation scenario for tokamaks with reasonable amount of computational power.

6. Conclusions and Perspectives

The models that describe the evolution of plasma in tokamak devices are highly non-linear. Alone the direct simulation, meaning a simulation that shows only how the plasma will evolve for certain given control inputs is a demanding task. Moreover, the corresponding inverse problem the much more relevant problem when

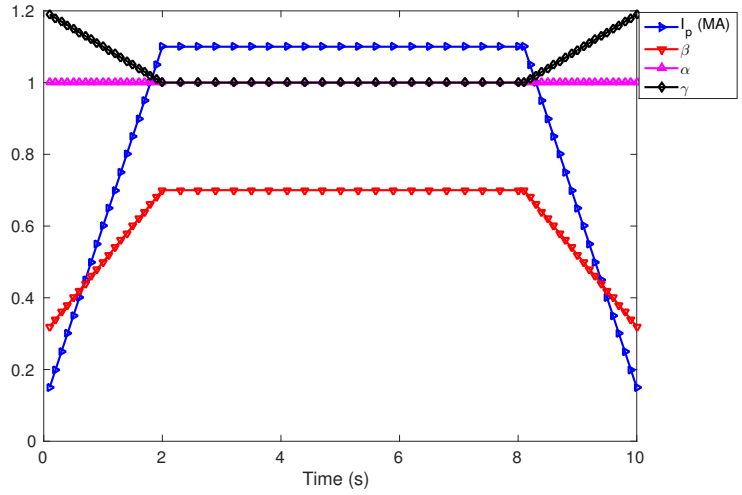


FIGURE 12. The prescribed I_p, and the parameters of the plasma profile for the full scenario simulation of HL-2M.

experimenting with tokamaks: what are the control parameters that ensure that the plasma evolves through certain prescribed states? From the engineering point of view it would be beneficial to provide the majority of control as feed-forward control that can be determined a priori, and to minimize the amount of feedback control that needs to be calculated during the operation. Until today, the calculation of the feed-forward control is mainly tackled by ODE models that have the flavor of electrical circuits [2]. These ODE models determine the coil voltages that correspond to coil currents of precomputed static equilibria. Particular feedback controllers are then necessary to ensure that the currents in the system are indeed close to the precomputed ones.

In this work we formulated an optimal control problem, that uses the force balance (4) and Maxwell's equations as constraints, to compute directly the optimal feed-forward control. The approach is more consistent, as it uses the full nonlinear model of partial differential equations. Then, in the solution of the optimization problem, it is Newton's method that determines consistently the linearization of this non-linear model.

We presented here the two basic ingredients, the finite element method and sequential quadratic programming, to set up such an optimal control approach for plasma evolution in tokamaks. The finite element method allows for an easy treatment of free plasma boundary and the decaying conditions at infinity. Further, we can, almost automatically, compute linearizations and the adjoint operators that are required for the optimization with sequential quadratic programming. We highlighted that existing implementations of Newton's methods for the direct simulation can be easily extended to an implementation of SQP-type methods for the corresponding inverse or optimal control problems. In the future we will have to augment the non-linear model (5)-(8) with further equations, namely the resistive diffusion and transport equations, that provide a more complete description.

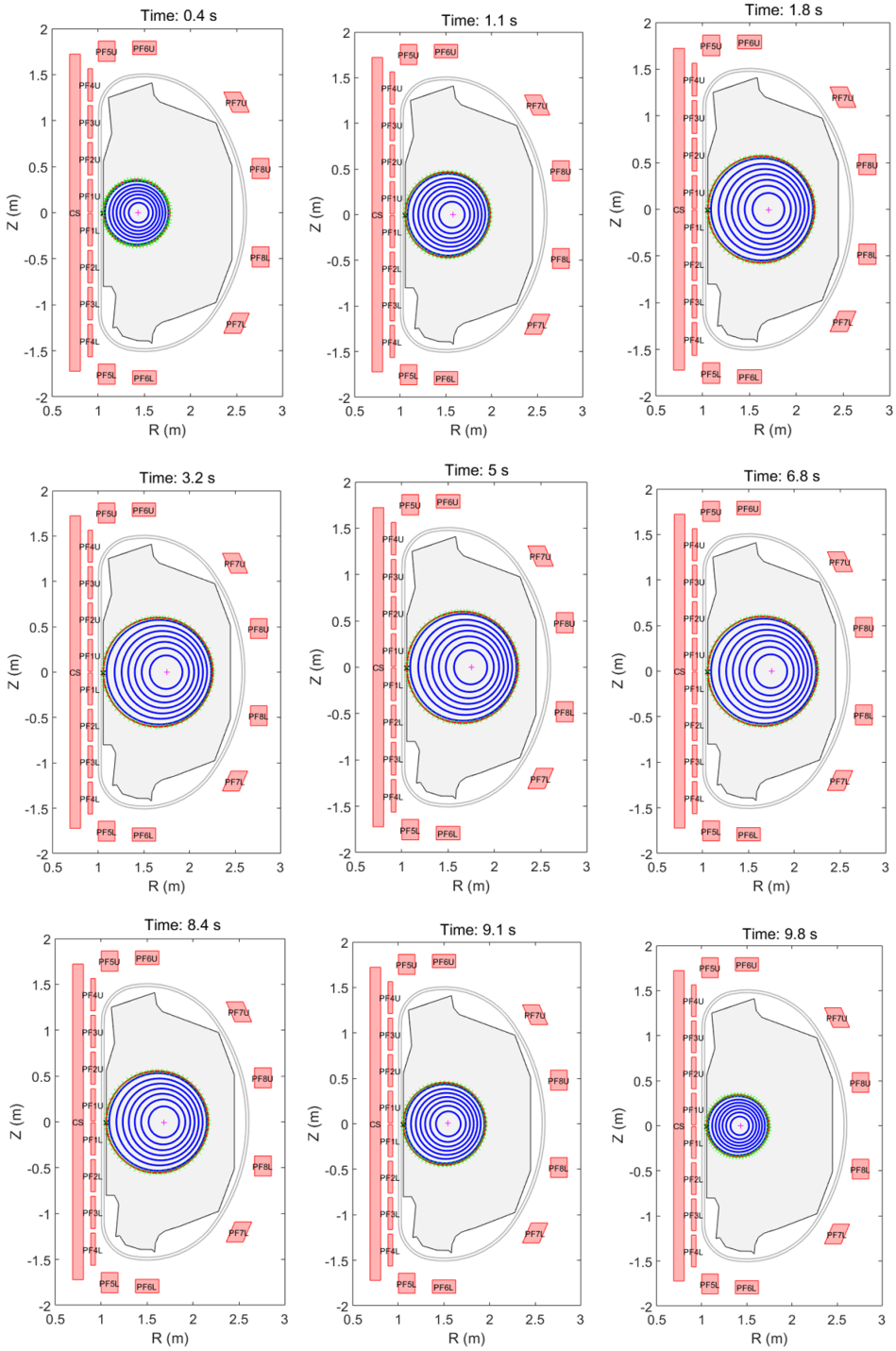


FIGURE 13. Level lines (blue) of the magnetic flux inside the plasma at selected times for the calculation of a full scenario (ramp up, flat top, ramp down) for HL-2M based on 60 timesteps with prescribed boundaries (green dots) at each time step.

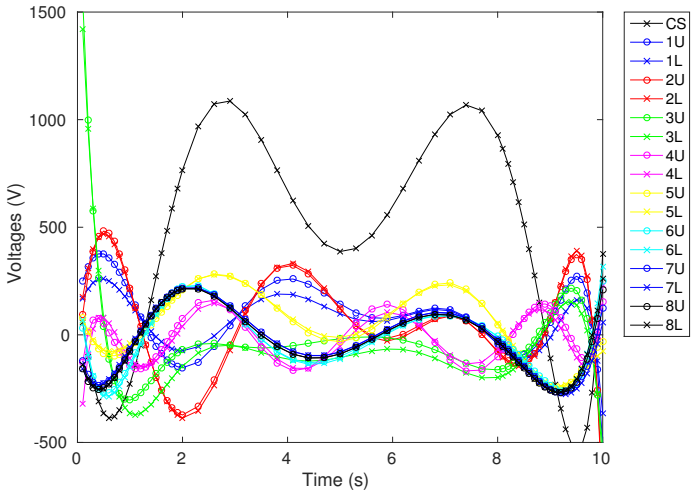


FIGURE 14. The voltages of the inverse evolution calculation for HL-2M.

APPENDIX A. FAST ALGORITHMS FOR OPTIMAL CONTROL PROBLEMS

The following section is a short summary on algorithms for general optimal control problems, where we focus on *finite dimensional optimal control problems*. This simplifies considerably the presentation, and is also more relevant for this work as we always work with discretized versions of (1). While most of the subsequent methods are well-known and can be found in excellent text books such as [15, 4, 11], we prefer to include this discussion to keep the presentation self-consistent.

We consider the following generic version of an optimal control problem

$$\min_{\mathbf{u},\mathbf{y}} J(\mathbf{y},\mathbf{u}) \quad \text{s.t.} \quad \mathbf{b}(\mathbf{y},\mathbf{u}) = \mathbf{0}\,, \tag{25}$$

where $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$ are the so-called state and control variables. The constraint $\mathbf{b}(\mathbf{y},\mathbf{u}) = \mathbf{0}$ with $\mathbf{b}(\mathbf{y},\mathbf{u}) \in \mathbb{R}^n$ is the discretization (in space and time) of (5), (8) and (9).

The optimization problem (25) is a constrained optimization problem that can be recast as an unconstrained optimization problem. For this we introduce the *reduced* objective function $\widehat{J}(\mathbf{u}) := J(\mathbf{y}(\mathbf{u}),\mathbf{u})$, with $\mathbf{y}(\mathbf{u})$ such that $\mathbf{b}(\mathbf{y}(\mathbf{u}),\mathbf{u}) = \mathbf{0}$. The gradient of $\widehat{J}(\mathbf{u})$ is

$$\widehat{J}_{\mathbf{u}}(\mathbf{u}) = J_{\mathbf{u}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) + J_{\mathbf{y}}(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{y}_{\mathbf{u}}(\mathbf{u})\,.$$

Some control $\mathbf{u}^*$ is called the stationary point of the reduced objective function if and only if

$$\widehat{J}_{\mathbf{u}}(\mathbf{u}^*) = \mathbf{0}\,. \tag{26}$$

An iterative approach to the solution of (26) faces two major challenges. First, in each iteration we have to solve the full non-linear problem $\mathbf{b}(\mathbf{y}(\mathbf{u}),\mathbf{u}) = \mathbf{0}$ for $\mathbf{y}(\mathbf{u})$ and second, we also need to compute the sensitivities $\mathbf{y}_{\mathbf{u}}(\mathbf{u})$. The second challenge

is addressed by the definition of the adjoint state $\mathbf{p}(\mathbf{u}) \in \mathbb{R}^n$ via

$$\mathbf{p}(\mathbf{u}) = -\mathbf{b}_{\mathbf{y}}^{-T}(\mathbf{y}(\mathbf{u}))J_{\mathbf{y}}^{T}(\mathbf{y}(\mathbf{u}),\mathbf{u})\,. \tag{27}$$

We see

$$\begin{aligned} 0 =& \mathbf{p}^T(\mathbf{u})\left(\mathbf{b}_{\mathbf{u}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) + \mathbf{b}_{\mathbf{y}}(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{y}_{\mathbf{u}}(\mathbf{u})\right) \\ =& \mathbf{p}^T(\mathbf{u})\mathbf{b}_{\mathbf{u}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) - J_{\mathbf{y}}(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{y}_{\mathbf{u}}(\mathbf{u})\,, \end{aligned}$$

so the gradient of $\widehat{J}(\mathbf{u})$ can also be expressed as

$$\widehat{J}_{\mathbf{u}}(\mathbf{u}) = J_{\mathbf{u}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) + \mathbf{p}^T(\mathbf{u})\mathbf{b}_{\mathbf{u}}(\mathbf{y}(\mathbf{u}),\mathbf{u})\,. \tag{28}$$

The Algorithm 2 sketches the gradient descent. Still, this algorithm requires in each

Algorithm 2 Gradient descent

$\Delta\mathbf{u} \leftarrow \mathbf{0}$, $\mathbf{y} \leftarrow \mathbf{y}^0$, $\mathbf{u} \leftarrow \mathbf{u}^0$
while $\|\Delta\mathbf{u}\| > 0$ **do**
  $\Delta\mathbf{y} \leftarrow \mathbf{0}$
  while $\|\Delta\mathbf{y}\| > 0$ **do**
    $\Delta\mathbf{y} \leftarrow -\mathbf{b}_{\mathbf{y}}^{-1}(\mathbf{y},\mathbf{u})\mathbf{b}(\mathbf{y},\mathbf{u})$
    $\mathbf{y} \leftarrow \mathbf{y} + \Delta\mathbf{y}$
  end while
  $\mathbf{p} \leftarrow -\mathbf{b}_{\mathbf{y}}^{-T}(\mathbf{y},\mathbf{u})J_{\mathbf{y}}^{T}(\mathbf{y},\mathbf{u})$
  $\Delta\mathbf{u} \leftarrow J_{\mathbf{u}}^{T}(\mathbf{y},\mathbf{u}) + \mathbf{b}_{\mathbf{u}}^{T}(\mathbf{y},\mathbf{u})\mathbf{p}$
  $\mathbf{u} \leftarrow \mathbf{u} + \Delta\mathbf{u}$
end while

iteration the solution of the non-linear constraint problem $\mathbf{b}(\mathbf{y}(\mathbf{u}),\mathbf{u}) = \mathbf{0}$ and the solution of the linear adjoint problem (27) for the evaluation of the reduced gradient $\widehat{J}(\mathbf{u})$. Moreover, the speed of convergence of the gradient descent algorithm is very slow. Fast convergence could be achieved in including second order terms, e.g. the Hessian $\widehat{J}_{\mathbf{uu}}(\mathbf{u})$ of the reduced objective function:

$$\begin{aligned} \widehat{J}_{\mathbf{uu}}(\mathbf{u}) =& \mathbf{Z}^T(\mathbf{u}) \begin{pmatrix} J_{\mathbf{yy}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) & J_{\mathbf{yu}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) \\ J_{\mathbf{uy}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) & J_{\mathbf{uu}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) \end{pmatrix} \mathbf{Z}(\mathbf{u}) \\ &+ \mathbf{Z}^T(\mathbf{u}) \begin{pmatrix} \mathbf{b}_{\mathbf{yy}}^T(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{p}(\mathbf{u}) & \mathbf{b}_{\mathbf{yu}}^T(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{p}(\mathbf{u}) \\ \mathbf{b}_{\mathbf{uy}}^T(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{p}(\mathbf{u}) & \mathbf{b}_{\mathbf{uu}}^T(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{p}(\mathbf{u}) \end{pmatrix} \mathbf{Z}(\mathbf{u}) \end{aligned} \tag{29}$$

with

$$\mathbf{Z}(\mathbf{u}) = \begin{pmatrix} -\mathbf{b}_{\mathbf{y}}^{-1}(\mathbf{y}(\mathbf{u}),\mathbf{u})\mathbf{b}_{\mathbf{u}}(\mathbf{y}(\mathbf{u}),\mathbf{u}) \\ \mathbf{1} \end{pmatrix},$$

but the requirement of solving repetitively the non-linear constraints $\mathbf{b}(\mathbf{y}(\mathbf{u}),\mathbf{u}) = \mathbf{0}$ remains a big drawback.

Sequential Quadratic Programming (SQP) is an entirely different approach that avoids this drawback and incorporates at the same time second order information. SQP is one of the most effective methods for non-linear constrained optimization with significant non-linearities in the constraints [15, Chapter 18]. To motivate SQP we recall that the previous discussion shows that the control $\mathbf{u}^*$ is a stationary point

of the reduced objective function if and only if there exist states $\mathbf{y}^*$ and adjoint states $\mathbf{p}^*$ such that

$$
\begin{aligned}
J_{\mathbf{y}}^T(\mathbf{y}^*,\mathbf{u}^*) + \mathbf{b}_{\mathbf{y}}^T(\mathbf{y}^*,\mathbf{u}^*)\mathbf{p}^* &= \mathbf{0}\,,\\
J_{\mathbf{u}}^T(\mathbf{y}^*,\mathbf{u}^*) + \mathbf{b}_{\mathbf{u}}^T(\mathbf{y}^*,\mathbf{u}^*)\mathbf{p}^* &= \mathbf{0}\,,\\
\mathbf{b}(\mathbf{y}^*,\mathbf{u}^*) &= \mathbf{0}\,,
\end{aligned}
\tag{30}
$$

A Newton-type method for solving (30) are iterations of the type

$$
\begin{pmatrix}
\mathbf{H}_{\mathbf{yy}}^k & \mathbf{H}_{\mathbf{yu}}^k & \mathbf{b}_{\mathbf{y}}^T(\mathbf{y}^k,\mathbf{u}^k)\\
\mathbf{H}_{\mathbf{uy}}^k & \mathbf{H}_{\mathbf{uu}}^k & \mathbf{b}_{\mathbf{u}}^T(\mathbf{y}^k,\mathbf{u}^k)\\
\mathbf{b}_{\mathbf{y}}(\mathbf{y}^k,\mathbf{u}^k) & \mathbf{b}_{\mathbf{u}}(\mathbf{y}^k,\mathbf{u}^k) & \mathbf{0}
\end{pmatrix}
\begin{pmatrix}
\mathbf{y}^{k+1}-\mathbf{y}^k\\
\mathbf{u}^{k+1}-\mathbf{u}^k\\
\mathbf{p}^{k+1}
\end{pmatrix}
= -
\begin{pmatrix}
J_{\mathbf{y}}^T(\mathbf{y}^k,\mathbf{u}^k)\\
J_{\mathbf{u}}^T(\mathbf{y}^k,\mathbf{u}^k)\\
\mathbf{b}(\mathbf{y}^k,\mathbf{u}^k)
\end{pmatrix}
\tag{31}
$$

with

$$
\begin{aligned}
\begin{pmatrix}
\mathbf{H}_{\mathbf{yy}}^k & \mathbf{H}_{\mathbf{yu}}^k\\
\mathbf{H}_{\mathbf{uy}}^k & \mathbf{H}_{\mathbf{uu}}^k
\end{pmatrix}
&:=
\begin{pmatrix}
\mathbf{H}_{\mathbf{yy}}(\mathbf{y}^k,\mathbf{u}^k,\mathbf{p}^k) & \mathbf{H}_{\mathbf{yu}}(\mathbf{y}^k,\mathbf{u}^k,\mathbf{p}^k)\\
\mathbf{H}_{\mathbf{uy}}(\mathbf{y}^k,\mathbf{u}^k,\mathbf{p}^k) & \mathbf{H}_{\mathbf{uu}}(\mathbf{y}^k,\mathbf{u}^k,\mathbf{p}^k)
\end{pmatrix}
:=\\
&\begin{pmatrix}
J_{\mathbf{yy}}(\mathbf{y}^k,\mathbf{u}^k) & J_{\mathbf{yu}}(\mathbf{y}^k,\mathbf{u}^k)\\
J_{\mathbf{uy}}(\mathbf{y}^k,\mathbf{u}^k) & J_{\mathbf{uu}}(\mathbf{y}^k,\mathbf{u}^k)
\end{pmatrix}
+
\begin{pmatrix}
\mathbf{b}_{\mathbf{yy}}^T(\mathbf{y}^k,\mathbf{u}^k)\mathbf{p}^k & \mathbf{b}_{\mathbf{yu}}^T(\mathbf{y}^k,\mathbf{u}^k)\mathbf{p}^k\\
\mathbf{b}_{\mathbf{uy}}^T(\mathbf{y}^k,\mathbf{u}^k)\mathbf{p}^k & \mathbf{b}_{\mathbf{uu}}^T(\mathbf{y}^k,\mathbf{u}^k)\mathbf{p}^k
\end{pmatrix}\,,
\end{aligned}
$$

which corresponds to the following quadratic optimization problem with linear constraints:

$$
\begin{aligned}
\min_{\mathbf{u}^{k+1},\mathbf{y}^{k+1}} &\left(\frac{1}{2}\begin{pmatrix}\mathbf{y}^{k+1}-\mathbf{y}^k\\ \mathbf{u}^{k+1}-\mathbf{u}^k\end{pmatrix}^T
\begin{pmatrix}\mathbf{H}_{\mathbf{yy}}^k & \mathbf{H}_{\mathbf{yu}}^k\\ \mathbf{H}_{\mathbf{uy}}^k & \mathbf{H}_{\mathbf{uu}}^k\end{pmatrix}
+
\begin{pmatrix}J_{\mathbf{y}}^T(\mathbf{y}^k,\mathbf{u}^k)\\ J_{\mathbf{u}}^T(\mathbf{y}^k,\mathbf{u}^k)\end{pmatrix}\right)
\begin{pmatrix}\mathbf{y}^{k+1}-\mathbf{y}^k\\ \mathbf{u}^{k+1}-\mathbf{u}^k\end{pmatrix}\\
\text{s.t}\quad &\mathbf{b}(\mathbf{y}^k,\mathbf{u}^k) + \mathbf{b}_{\mathbf{y}}(\mathbf{y}^k,\mathbf{u}^k)(\mathbf{y}^{k+1}-\mathbf{y}^k) + \mathbf{b}_{\mathbf{u}}(\mathbf{y}^k,\mathbf{u}^k)(\mathbf{u}^{k+1}-\mathbf{u}^k) = \mathbf{0}\,.
\end{aligned}
$$

This sequence of quadratic optimization problems is at the origin of the name *sequential quadratic programming*.

If the linear systems in (31) become too large, it is the common practice to pursue the *null space approach* to arrive at the SQP formulation with the reduced Hessian. In introducing

$$
\mathbf{Z}_k = \begin{pmatrix}-\mathbf{b}_{\mathbf{y}}^{-1}(\mathbf{y}^k,\mathbf{u}^k)\mathbf{b}_{\mathbf{u}}(\mathbf{y}^k,\mathbf{u}^k)\\ \mathbf{1}\end{pmatrix}
\quad\text{and}\quad
\mathbf{Y}_k = \begin{pmatrix}-\mathbf{b}_{\mathbf{y}}^{-1}(\mathbf{y}^k,\mathbf{u}^k)\\ \mathbf{0}\end{pmatrix}
\tag{32}
$$

we obtain the identity

$$
\begin{pmatrix}\mathbf{y}^{k+1}-\mathbf{y}^k\\ \mathbf{u}^{k+1}-\mathbf{u}^k\end{pmatrix} = \mathbf{Z}_k(\mathbf{u}^{k+1}-\mathbf{u}^k) + \mathbf{Y}_k\mathbf{b}(\mathbf{y}^k,\mathbf{u}^k)
\tag{33}
$$

and find the following linear system for the increment $\mathbf{u}^{k+1}-\mathbf{u}^k$

$$
\mathbf{M}(\mathbf{y}^k,\mathbf{u}^k)(\mathbf{u}^{k+1}-\mathbf{u}^k) = -\mathbf{m}(\mathbf{y}^k,\mathbf{u}^k)
\tag{34}
$$

with

$$
\mathbf{M}(\mathbf{y}^k,\mathbf{u}^k) := \mathbf{Z}_k^T\begin{pmatrix}\mathbf{H}_{\mathbf{yy}}^k & \mathbf{H}_{\mathbf{yu}}^k\\ \mathbf{H}_{\mathbf{uy}}^k & \mathbf{H}_{\mathbf{uu}}^k\end{pmatrix}\mathbf{Z}_k
$$

and

$$
\mathbf{m}(\mathbf{y}^k,\mathbf{u}^k) := \mathbf{Z}_k^T\left(\begin{pmatrix}J_{\mathbf{y}}^T(\mathbf{y}^k,\mathbf{u}^k)\\ J_{\mathbf{u}}^T(\mathbf{y}^k,\mathbf{u}^k)\end{pmatrix} + \begin{pmatrix}\mathbf{H}_{\mathbf{yy}}^k & \mathbf{H}_{\mathbf{yu}}^k\\ \mathbf{H}_{\mathbf{uy}}^k & \mathbf{H}_{\mathbf{uu}}^k\end{pmatrix}\mathbf{Y}_k\mathbf{b}(\mathbf{y}^k,\mathbf{u}^k)\right).
$$

It is insightful to compare the expressions involved in the reduced formulation (34) of SQP to the gradient (28) and the Hessian (29) of the reduced objective function: the gradient and Hessian of the reduced objective are equal to $\mathbf{m}(\mathbf{y},\mathbf{u})$ and $\mathbf{M}(\mathbf{y},\mathbf{u})$ only when the state $\mathbf{y}$ and control $\mathbf{u}$ verify the non-linear direct problem.

In general, iterative methods, such as the conjugate gradient (CG) methods, are used to solve (34). Within each iteration step two linear systems corresponding to $\mathbf{b}_\mathbf{y}^{-1}(\mathbf{y}^k, \mathbf{u}^k)$ and $\mathbf{b}_\mathbf{y}^{-T}(\mathbf{y}^k, \mathbf{u}^k)$ (see Algorithm 3) need to be inverted. The CG algorithm (Algorithm 3) appears different than standard formulations, as we update within the CG-iterations not only the control but also the state unknown. This avoids solving one additional non-linear direct problem after each CG call.

Algorithm 3 SQP with CG iterative solver (less memory intensive)

$\Delta\mathbf{u} \leftarrow 1$, $\Delta\mathbf{y} \leftarrow 1$, $\mathbf{y} \leftarrow \mathbf{y}^0$, $\mathbf{u} \leftarrow \mathbf{u}^0$, $\mathbf{p} \leftarrow \mathbf{p}^0$
while $\|\Delta\mathbf{u}\| > 0$, $\|\Delta\mathbf{y}\| > 0$ **do**
  $\Delta\mathbf{u} \leftarrow 0$, $\Delta\mathbf{y} \leftarrow -\mathbf{b}_\mathbf{y}^{-1}(\mathbf{y},\mathbf{u})\mathbf{b}(\mathbf{y},\mathbf{u})$
  $\Delta\mathbf{p} \leftarrow -\mathbf{b}_\mathbf{y}^{-T}(\mathbf{y},\mathbf{u})(J_\mathbf{y}^T(\mathbf{y},\mathbf{u}) + \mathbf{H}_{\mathbf{yy}}(\mathbf{y},\mathbf{u},\mathbf{p})\Delta\mathbf{y})$
  $\mathbf{r} \leftarrow J_\mathbf{u}^T(\mathbf{y},\mathbf{u}) + \mathbf{H}_{\mathbf{uy}}(\mathbf{y},\mathbf{u},\mathbf{p})\Delta\mathbf{y} + \mathbf{b}_\mathbf{u}^T(\mathbf{y},\mathbf{u})\Delta\mathbf{p}$
  $\mathbf{s} \leftarrow -\mathbf{r}$
  while $\|\mathbf{r}\| > 0$ **do**
    $\mathbf{a} \leftarrow -\mathbf{b}_\mathbf{y}^{-1}(\mathbf{y},\mathbf{u})\mathbf{b}_\mathbf{u}(\mathbf{y},\mathbf{u})\mathbf{s}$
    $\mathbf{b} \leftarrow -\mathbf{b}_\mathbf{y}^{-T}(\mathbf{y},\mathbf{u})(\mathbf{H}_{\mathbf{yu}}(\mathbf{y},\mathbf{u},\mathbf{p})\mathbf{s} + \mathbf{H}_{\mathbf{yy}}(\mathbf{y},\mathbf{u},\mathbf{p})\mathbf{a})$
    $\widetilde{\mathbf{s}} \leftarrow \mathbf{H}_{\mathbf{uu}}(\mathbf{y},\mathbf{u},\mathbf{p})\mathbf{s} + \mathbf{H}_{\mathbf{uy}}(\mathbf{y},\mathbf{u},\mathbf{p})\mathbf{a} + \mathbf{b}_\mathbf{u}^T(\mathbf{y},\mathbf{u})\mathbf{b}$
    $\alpha \leftarrow \frac{\mathbf{r}^T\mathbf{r}}{\mathbf{s}^T\widetilde{\mathbf{s}}}$
    $\Delta\mathbf{u} \leftarrow \Delta\mathbf{u} + \alpha\mathbf{s}$, $\Delta\mathbf{y} \leftarrow \Delta\mathbf{y} + \alpha\mathbf{a}$, $\Delta\mathbf{p} \leftarrow \Delta\mathbf{p} + \alpha\mathbf{b}$
    $\widetilde{\mathbf{r}} \leftarrow \mathbf{r} + \alpha\widetilde{\mathbf{s}}$
    $\beta \leftarrow \frac{\widetilde{\mathbf{r}}^T\widetilde{\mathbf{r}}}{\mathbf{r}^T\mathbf{r}}$
    $\mathbf{s} \leftarrow -\widetilde{\mathbf{r}} + \beta\mathbf{s}$
    $\mathbf{r} \leftarrow \widetilde{\mathbf{r}}$
  end while
  $\mathbf{u} \leftarrow \mathbf{u} + \Delta\mathbf{u}$, $\mathbf{y} \leftarrow \mathbf{y} + \Delta\mathbf{y}$, $\mathbf{p} \leftarrow \Delta\mathbf{p}$
end while

Alternatively, if we have sufficient memory to store $\mathbf{M}(\cdot,\cdot)$, we can compute $\mathbf{M}(\cdot,\cdot)$ explicitly. We never compute neither $\mathbf{b}_\mathbf{y}^{-1}(\mathbf{y}^k, \mathbf{u}^k)$ nor $\mathbf{b}_\mathbf{y}^{-T}(\mathbf{y}^k, \mathbf{u}^k)$ explicitly. This alternative approach is summarized in Algorithm 4.

Algorithm 4 SQP with direct solver (memory intensive)

$\Delta\mathbf{u} \leftarrow 1$, $\Delta\mathbf{y} \leftarrow 1$, $\mathbf{y} \leftarrow \mathbf{y}^0$, $\mathbf{u} \leftarrow \mathbf{u}^0$, $\mathbf{p} \leftarrow \mathbf{p}^0$
while $\|\Delta\mathbf{u}\| > 0$, $\|\Delta\mathbf{y}\| > 0$ **do**
  $(\Delta\mathbf{y}, \mathbf{Y}) \leftarrow -\mathbf{b}_\mathbf{y}^{-1}(\mathbf{y},\mathbf{u})(\mathbf{b}(\mathbf{y},\mathbf{u}), \mathbf{b}_\mathbf{u}(\mathbf{y},\mathbf{u}))$
  $\mathbf{m} \leftarrow J_\mathbf{u}^T(\mathbf{y},\mathbf{u}) + \mathbf{Y}^T J_\mathbf{y}^T(\mathbf{y},\mathbf{u}) + \mathbf{H}_{\mathbf{uy}}(\mathbf{y},\mathbf{u},\mathbf{p})\Delta\mathbf{y} + \mathbf{Y}^T\mathbf{H}_{\mathbf{yy}}(\mathbf{y},\mathbf{u},\mathbf{p})\Delta\mathbf{y}$
  $\mathbf{M} \leftarrow \mathbf{H}_{\mathbf{uu}}(\mathbf{y},\mathbf{u},\mathbf{p}) + \mathbf{Y}^T\mathbf{H}_{\mathbf{yu}}(\mathbf{y},\mathbf{u},\mathbf{p}) + \mathbf{H}_{\mathbf{uy}}(\mathbf{y},\mathbf{u},\mathbf{p})\mathbf{Y} + \mathbf{Y}^T\mathbf{H}_{\mathbf{yy}}(\mathbf{y},\mathbf{u},\mathbf{p})\mathbf{Y}$
  $\Delta\mathbf{u} \leftarrow -\mathbf{M}^{-1}\mathbf{m}$
  $\Delta\mathbf{y} \leftarrow \Delta\mathbf{y} + \mathbf{Y}\Delta\mathbf{u}$
  $\mathbf{p} \leftarrow -\mathbf{b}_\mathbf{y}^{-T}(\mathbf{y},\mathbf{u})(J_\mathbf{y}^T(\mathbf{y},\mathbf{u}) + \mathbf{H}_{\mathbf{yy}}(\mathbf{y},\mathbf{u},\mathbf{p})\Delta\mathbf{y} + \mathbf{H}_{\mathbf{yu}}(\mathbf{y},\mathbf{u},\mathbf{p})\Delta\mathbf{u})$
  $\mathbf{u} \leftarrow \mathbf{u} + \Delta\mathbf{u}$
  $\mathbf{y} \leftarrow \mathbf{y} + \Delta\mathbf{y}$
end while

Appendix B. Circuit Equations

The external circuit of poloidal field coils and suppliers is represented by a directed graph with nodes and directed edges between nodes. We assign to each directed edge $\mathbf{s} = (i, j)$ between two nodes with index i and j a (directed) voltage $V_{\mathbf{s}}$ and current $I_{\mathbf{s}}$. For an edge $\mathbf{s}$ representing a coil $\mathcal{C}$ we have

$$I_{\mathbf{s}} = \mathbf{Z}_{\mathbf{ss}} V_{\mathbf{s}} + (\vec{F}_C(\frac{d}{dt}\psi))_{\mathbf{s}} := \frac{n_{\mathbf{s}}}{R_{\mathbf{s}}} V_{\mathbf{s}} - 2\pi \frac{n_{\mathbf{s}}^2}{R_{\mathbf{s}}} \frac{1}{\mathcal{C}} \int_{\mathcal{C}} \frac{d}{dt}\psi dr dz \,, \tag{35}$$

where $R_{\mathbf{s}}$ and $n_{\mathbf{s}}$ are the total resistance and the wire turns of the coil.

Then we introduce the incidence matrix $\mathbf{G}$ that has entry $\mathbf{G}_{\mathbf{s},i} = 0$ when the node with index i is not contained in the edge with index $\mathbf{s} = (i, j)$, or entry $\mathbf{G}_{\mathbf{s},i} = 1(-1)$ when the node with index i is contained in the edge with index $\mathbf{s} = (i, j)$ and induced orientation coincides (coincides not) with the orientation of the edge. Likewise we can introduce oriented polygons, whose boundaries are the edges of the graph and an incidence matrix $\mathbf{C}$ for edges and polygons. Then we have $\mathbf{CG} = 0$ and the Kirchhoff current and voltage laws are:

$$\mathbf{G}^T \vec{I} = 0 \quad \text{and} \quad \mathbf{C}\vec{V} = 0 \,,$$

where the components of $\vec{V}$ and $\vec{I}$ are the voltages and currents associated to the edges.

In the following we assume that the edges of the circuit correspond either to a coil or to an external voltage supplier/source. We introduce the subscripts $_S$ and $_C$ to distinguish between edges corresponding to supplies and edges corresponding to coils, e.g. $\mathbf{G}^T\vec{I} = \mathbf{G}_S^T\vec{I}_S + \mathbf{G}_C^T\vec{I}_C = 0$. Moreover we introduce a (node) potential $\vec{U}$ with $\vec{V} = \mathbf{G}\vec{U}$ and hence $\vec{V}_S = \mathbf{G}_S\vec{U}$ and $\vec{V}_C = \mathbf{G}_C\vec{U}$. In summary we arrive at:

$$\begin{pmatrix} -\mathbf{Z}^{-1} & 0 & \mathbf{G}_C \\ 0 & 0 & \mathbf{G}_S \\ \mathbf{G}_C^T & \mathbf{G}_S^T & 0 \end{pmatrix} \begin{pmatrix} \vec{I}_C \\ \vec{I}_S \\ \vec{U} \end{pmatrix} = \begin{pmatrix} -\mathbf{Z}^{-1}\vec{F}_C(\frac{d}{dt}\psi) \\ \vec{V}_S \\ 0 \end{pmatrix} . \tag{36}$$

or, to be close with the notation from CEDRES++,

$$\begin{pmatrix} -\mathbf{Z}^{-1} & 0 \\ 0 & 0 \\ \mathbf{G}_C^T & \mathbf{G}_S^T \end{pmatrix} \begin{pmatrix} \vec{I}_C \\ \vec{I}_S \end{pmatrix} + \begin{pmatrix} \mathbf{G}_C \\ \mathbf{G}_S \\ 0 \end{pmatrix} (\vec{U}) = \begin{pmatrix} 0 \\ \vec{V}_S \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{Z}^{-1}\vec{F}_C(\frac{d}{dt}\psi) \\ 0 \\ 0 \end{pmatrix} . \tag{37}$$

Well-posedness requires to fix $\vec{U}$ at one node and to remove the Kirchoff current equation for the same node. We choose an arbitrary node and fix $\vec{U}$ to zero at this point. This is equivalent to cancel a column (row) in $\mathbf{G}$ ($\mathbf{G}^T$), hence in the subsequent lines $\mathbf{G}_\cdot$ ($\mathbf{G}_\cdot^T$) always refers to the reduced matrices.

After some tedious calculations we find

$$\begin{pmatrix} \vec{I}_C \\ \vec{I}_S \\ \vec{U} \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{S} & * \\ * & * & * \\ \mathbf{Q} & \mathbf{T} & * \end{pmatrix} \begin{pmatrix} \vec{F}_C(\frac{d}{dt}\psi) \\ \vec{V}_S \\ 0 \end{pmatrix} , \tag{38}$$

with

$$\mathbf{R} = \mathbf{I} + \mathbf{Z}\mathbf{G}_C\mathbf{K}\mathbf{G}_C^T + \mathbf{Z}\mathbf{G}_C\mathbf{K}\mathbf{G}_S^T\mathbf{M}\mathbf{G}_S\mathbf{K}\mathbf{G}_C^T$$
$$\mathbf{Q} = \mathbf{G}_C\mathbf{K}\mathbf{G}_C^T + \mathbf{G}_C\mathbf{K}\mathbf{G}_S^T\mathbf{M}\mathbf{G}_S\mathbf{K}\mathbf{G}_C^T$$

and

$$\mathbf{S} = -\mathbf{Z}\mathbf{G}_C\mathbf{K}\mathbf{G}_S^T\mathbf{M}\,, \qquad \mathbf{T} \;\; = -\mathbf{G}_C\mathbf{K}\mathbf{G}_S^T\mathbf{M}\,,$$

where $\mathbf{K}^{-1} = -\mathbf{G}_C^T\mathbf{Z}\mathbf{G}_C$, and $\mathbf{M}^{-1} = -\mathbf{G}_S\mathbf{K}\mathbf{G}_S^T$.

References

[1] R. Albanese, J. Blum, and O. Barbieri. On the solution of the magnetic flux equation in an infinite domain. In *EPS. 8th Europhysics Conference on Computing in Plasma Physics (1986)*, pages 41–44, 1986.

[2] M. Ariola and A. Pironti. *Magnetic Control of Tokamak Plasmas*. Springer London, 2008.

[3] J. Blum. *Numerical simulation and optimal control in plasma physics*. Wiley/Gauthier-Villars, 1989.

[4] J.F. Bonnans. *Numerical Optimization: theoretical and practical aspects : with 26 figures*. Universitext (Berlin. Print). Springer, 2003.

[5] C. Bourdelle, J.F. Artaud, V. Basiuk, M. Bécoulet, S. Brémond, J. Bucalossi, H. Bufferand, G. Ciraolo, L. Colas, Y. Corre, et al. West physics basis. *Nuclear Fusion*, 55(6):063017, 2015.

[6] A. Fasoli, S. Brunner, W. A. Cooper, J. P. Graves, P. Ricci, O. Sauter, and L. Villard. Computational challenges in magnetic-confinement fusion physics. *Nat Phys*, 12(5):411–423, May 2016.

[7] B. Faugeras and H. Heumann. Fem-bem coupling methods for tokamak plasma axisymmetric free-boundary equilibrium computations in unbounded domains. *Journal of Computational Physics*, pages –, 2017.

[8] H. Grad and J. Hogan. Classical diffusion in a tokamak. *Phys. Rev. Lett.*, 24:1337–1340, Jun 1970.

[9] V. Grandgirard. *Modélisation de l'équilibre d'un plasma de tokamak*. PhD thesis, l'Université de Franche-Comté, 1999.

[10] H. Heumann, J. Blum, C. Boulbe, B. Faugeras, G. Selig, J.-M. Ané, S. Brémond, V. Grandgirard, P. Hertout, and E. Nardon. Quasi-static free-boundary equilibrium of toroidal plasma with CEDRES++: Computational methods and applications. *Journal of Plasma Physics*, 81, 6 2015.

[11] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer, New York, 2009.

[12] S. C. Jardin. *Computational methods in plasma physics*. Boca Raton, FL : CRC Press/Taylor & Francis, 2010.

[13] Qiang Li et al. The component development status of HL-2M tokamak. *Fusion Engineering and Design*, 96:338–342, 2015.

[14] E. Nardon, H. Heumann, J.F. Artaud, J. Bucalossi, D. Douai, N. Fedorczak, T. Loarer, P. Moreau, R. Nouailletas, C. Reux, F. Saint-Laurent, P. Tamain, and the WEST team. Magnetic configuration and plasma start-up in the west tokamak. In *45th EPS Conference on Plasma Physics*, 2018.

[15] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[16] J. Ongena, R. Koch, R. Wolf, and H. Zohm. Magnetic-confinement fusion. *Nat Phys*, 12(5):398–410, May 2016.

[17] J. Wesson. *Tokamaks*. The International Series of Monographs in Physics. Oxford University Press, 2004.

E-mail address: jblum@math.unice.fr

Université Côte d'Azur, Inria, CNRS, Laboratoire Jean-Alexandre Dieudonné (LJAD), Parc Valrose, 06108 Nice Cedex02 France

E-mail address: holger.heumann@inria.fr

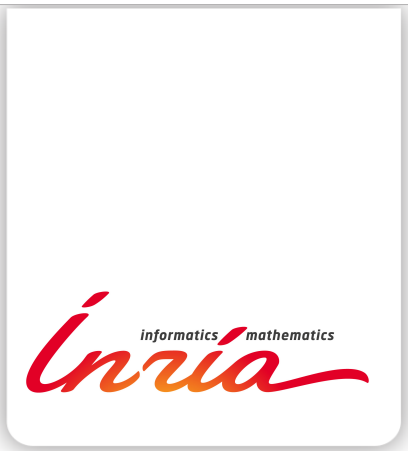
Université Côte d'Azur, Inria, CNRS, Laboratoire Jean-Alexandre Dieudonné (LJAD), Parc Valrose, 06108 Nice Cedex02 France

E-mail address: eric.nardon@cea.fr

CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France

E-mail address: xiao.song@inria.fr

Université Côte d'Azur, Inria, CNRS, Laboratoire Jean-Alexandre Dieudonné (LJAD), Parc Valrose, 06108 Nice Cedex02 France & Southwestern Institute of Physics, P.O.Box 432, Chengdu, 610041, China



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399