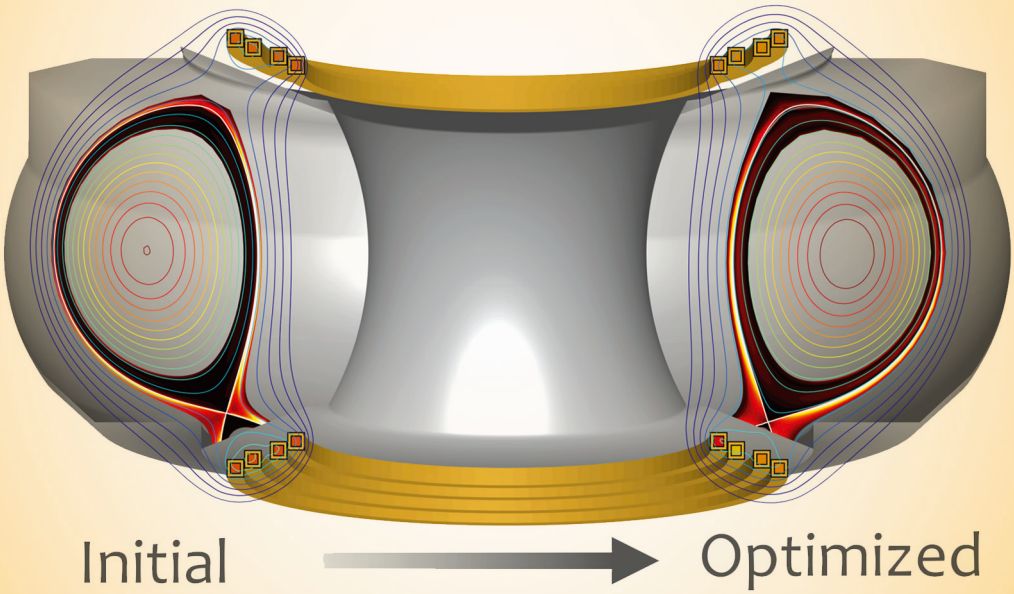
Automated Magnetic Divertor Design for Optimal Power Exhaust

Maarten Blommaert



Energie & Umwelt / Energy & Environment
Band / Volume 365
ISBN 978-3-95806-216-0

JÜLICH
FORSCHUNGSZENTRUM

Schriften des Forschungszentrums Jülich
Reihe Energie & Umwelt / Energy & Environment Band / Volume 365

Forschungszentrum Jülich GmbH
Institute of Energy and Climate Research
Plasma Physics IEK-4

Automated Magnetic Divertor Design for Optimal Power Exhaust

Maarten Blommaert

Schriften des Forschungszentrums Jülich
Reihe Energie & Umwelt / Energy & Environment Band / Volume 365

ISSN 1866-1793 ISBN 978-3-95806-216-0

Bibliographic information published by the Deutsche Nationalbibliothek.
The Deutsche Nationalbibliothek lists this publication in the Deutsche
Nationalbibliografie; detailed bibliographic data are available in the
Internet at http://dnb.d-nb.de.

Publisher and Distributor: Forschungszentrum Jülich GmbH
Zentralbibliothek
52425 Jülich
Tel: +49 2461 61-5368
Fax: +49 2461 61-6103
Email: zb-publikation@fz-juelich.de
www.fz-juelich.de/zb

Cover Design: Grafische Medien, Forschungszentrum Jülich GmbH

Printer: Grafische Medien, Forschungszentrum Jülich GmbH

Schriften des Forschungszentrums Jülich
Reihe Energie & Umwelt / Energy & Environment, Band / Volume 365

D 82 (Diss. RWTH Aachen University, 2016)

ISSN 1866-1793
ISBN 978-3-95806-216-0

The complete volume is freely available on the Internet on the Jülicher Open Access Server (JuSER)
at www.fz-juelich.de/zb/openaccess.

Preface

> When we look up at night and view the stars, everything we see is shining because of distant nuclear fusion.
>
> Carl Sagan

Harnessing fusion power on earth is an extremely challenging task. Yet, the potential impact on our future energy system is enormous. These two things make it an interesting puzzle to solve. Although the PhD research presented in this dissertation only contributes a very small piece to this puzzle, I truly enjoyed it and had much fun learning from others, trying new things, and seeing something finally work after a hundred attempts. During this trial-and-error process that—as I learned—is characteristic to research, I had the opportunity to visit and work in many inspiring environments. In this preface, I would like to seize the opportunity to thank some people in particular that undoubtedly influenced this work either directly, or indirectly.

Let me start by thanking the one person that is—in my opinion—wrongfully missing on the cover of this booklet. Tine (Baelmans)—as I learned to call you the last couple of years when around colleagues—thank you for all your guidance, for helping me to structure my research, for interesting discussions in the car on our way to Jülich or Brasschaat, and—most of all—for being a great mum!

> Science is built up of facts, as a house is built of stones but an accumulation of facts is no more a science than a heap of stones is a house.
>
> Henri Poincaré

I also owe a debt of gratitude to my supervisors Nicolas Gauger and Detlev Reiter. To Nico, for introducing me to *real* one-shot optimization and even more so for the hospitable welcomes in both Aachen and Kaiserslautern. To Detlev, for sharing his physics insights, for contextualizing this research in the bigger field of fusion research, and for advocating this optimal design research in the fusion community. I thank the Forschungszentrum Jülich (FZJ) for the financial support.

I would like to thank my examination committee for taking the time to review the PhD research and text, and for their genuine interest in this research. I thank Michael Rack for providing a German translation of the abstract, and Lisa Kusch and Steffi Günther for proofreading pieces of text.

As for my colleagues in Jülich, Kaiserslautern, and Leuven, I appreciate your contribution to making these past couple of years an unforgettable period. Let me mention just a few of the people here that earn a special word of appreciation. Not knowing the German language and culture was initially a serious difficulty. Fortunately, I could always count on Petra Börner to help me out with German documents and all kinds of practical issues. Furthermore, the monthly Gyros-lunches with her, Tina, and Tine were always something to look forward to. Next, I want to thank Wouter Dekeyser for sharing his passion for nuclear fusion research with me. The long inspiring discussions we already had during my master thesis motivated me to take on this PhD research. I hope we can have many more! Ruben Gielen, thank you for teaching me the value of lean and aesthetical research presentations. Moreover, you earn the title of most *assistant* assistant I've encountered! Tijs van Oevelen, your scepticism is something beyond compare that is both educative and entertaining, and gave us some truly unforgettable discussions at the TME lunch table. Geert (Buckinx), although I admire the distinctive rigour that characterizes your research, you earned your place in this preface with the many evenings we spent philosophizing, joking, and drinking 'till

the early hours of the morning in Leuven's finest bars. A special mentioning also goes to all my colleagues at the TME division for the fun we had at informal and entertaining lunch breaks, TME activities, and weekends.

Also outside the workplace there are people I need to thank. My parents, brothers, and friends I thank for their support and beguilement when I needed it most. The final word of thanks goes to my beautiful girlfriend Elza, who has throughout my PhD research given me love, support, and a warm refuge that was like an infinite source of energy to me.

> Energy is the key to creativity. Energy is the key to life.
>
> William Shatner

Abstract

The so-called *divertor* is the standard particle and power exhaust system of nuclear fusion tokamaks. In essence, the magnetic configuration hereby 'diverts' the plasma to a specific divertor structure. The design of this divertor is still a key issue to be resolved to evolve from experimental fusion tokamaks to commercial power plants. The focus of this dissertation is on one particular design requirement: avoiding excessive heat loads on the divertor structure. The divertor design process is assisted by plasma edge transport codes that simulate the plasma and neutral particle transport in the edge of the reactor. These codes are computationally extremely demanding, not in the least due to the complex collisional processes between plasma and neutrals that lead to strong radiation sinks and macroscopic heat convection near the vessel walls.

One way of improving the heat exhaust is by modifying the magnetic confinement that governs the plasma flow. In this dissertation, automated design of the magnetic configuration is pursued using adjoint based optimization methods. A simple and fast perturbation model is used to compute the magnetic field in the vacuum vessel. A stable optimal design method of the nested type is then elaborated that strictly accounts for several nonlinear design constraints and code limitations. Using appropriate cost function definitions, the heat is spread more uniformly over the high-heat load plasma-facing components in a practical design example. Furthermore, practical in-parts adjoint sensitivity calculations are presented that provide a way to an efficient optimization procedure. Results are elaborated for a fictituous JET (Joint European Torus) case. The heat load is strongly

reduced by exploiting an expansion of the magnetic flux towards the solid divertor structure.

Subsequently, shortcomings of the perturbation model for magnetic field calculations are discussed in comparison to a *free boundary equilibrium* (FBE) simulation. These flaws in the magnetic model are then overcome by elaborating a strategy to include the full FBE code into the optimal design approach. Using the full model, results are then presented in application to the novel WEST (tungsten (W) Environment in Steady-state Tokamak) divertor.

Finally, *one-shot optimization* methods are considered for further acceleration of the optimal design procedure. Instead of fully solving state and adjoint equations in each optimization iteration, one-shot methods perform only a single iteration of state and adjoint solver in each optimization iteration. To reduce the cost of design updates, a grid deformation method is derived for strictly flux-aligned grids. Starting from a literature review, a novel one-shot strategy is then elaborated that features the globalization approach of state-of-the-art one-shot methods while yielding increased efficiency and practical usability. On an unconstrained test case, the novel method shows stable convergence.

Zusammenfassung

Der so genannte Divertor ist das übliche Abfuhrsystem für Teilchen und Leistung in einem Kernfusionsreaktor der Gattung Tokamak. Im Wesentlichen leitet die magnetische Konfiguration das Plasma zu einer bestimmten Divertor-Struktur um. Das Design dieses Divertors ist nach wie vor ein wichtiges zu lösendes Thema, um aus Tokamak Fusionsexperimenten kommerzielle Kraftwerken zu entwickeln. Der Schwerpunkt dieser Arbeit liegt auf einer bestimmten Designanforderung: Der Vermeidung übermäßiger Wärmelasten auf der Divertor-Struktur. Der Designprozess des Divertors wird durch Plasmarand-Transportcodes unterstützt, die den Plasma- und Neutralteilchentransport im Rand des Reaktors simulieren. Diese Codes sind rechnerisch extrem anspruchsvoll, vor allem aufgrund der komplexen Kollisionsprozesse zwischen Plasma- und Neutralteilchen, die zu starken Strahlungssenken und makroskopischer Wärmekonvektion in der Nähe der Gefäßwände führen.

Eine Möglichkeit, die Wärmeabfuhr zu verbessern, ist durch Modifizierung des magnetischen Einschlusses, der den Plasmafluss regelt. In dieser Dissertation wird das automatisierte Design der magnetischen Konfiguration basierend auf adjungierten Optimierungsverfahren untersucht. Ein einfaches und schnelles Störungsmodell wird verwendet, um das Magnetfeld in dem Vakuumbehälter zu berechnen. Eine stabile Methode des verschachelten Typs zur Ermittlung des optimalen Designs wird ausgearbeitet, die mehrere nicht-lineare Designeinschränkungen und Codelimitierungen berücksichtigt. Mit entsprechenden Definitionen der Kostenfunktion ist in einem praktischen Designbeispiel die Wärme gleichmäßiger über die stark wärme-

belasteten Wandkomponenten verteilt. Außerdem werden praktische teilweise adjungierte Sensitivitätsrechnungen vorgestellt, die eine Möglichkeit zu einem effizienten Optimierungsverfahren bieten. Die Ergebnisse sind für einen fiktiven JET (Joint European Torus) Fall ausgearbeitet. Die Wärmebelastung wird unter Verwendung einer Aufweitung des magnetischen Flusses hin zur Divertor-Struktur stark reduziert.

Anschließend werden Mängel des Störungsmodells für die magnetische Feldberechnungen im Vergleich zu einer Gleichgewichtssimulation mit freien Rändern (FBE) (Free-Boundary Equilibrium) diskutiert. Diese Mängel in dem magnetischen Modell werden überwunden, indem eine Strategie ausgearbeitet wird, die einen vollständigen FBE Code in den Optimierungsansatz integriert. Unter Verwendung des vollständigen Modells werden Ergebnisse einer Anwendung auf den neuen WEST-Divertor (tungsten (W) Environment in Steady-state Tokamak) präsentiert.

Abschließend werden One-Shot-Optimierungsverfahren für eine weitere Beschleunigung des Optimierungsansatz betrachtet. Statt Zustandsgleichungen und adjungierte Gleichungen in jeder Optimierungsiteration vollständig zu lösen wird beim One-Shot-Verfahren nur eine einzelne Iteration des Lösers für die Zustandsgleichung und adjungierte Gleichung pro Optimierungsiteration durchgeführt. Um die Kosten für Designanpassungen zu reduzieren wird eine Gitterdeformationsmethode für streng Fluss-ausgerichtete Gitter hergeleitet. Ausgehend von einer Literaturrecherche, wird eine neue One-Shot-Strategie erarbeitet, die den Globalisierungansatz der modernsten One-Shot-Verfahren besitzt, während gleichzeitig eine höhere Effizienz und praktische Verwendbarkeit gegeben ist. Bei einem unbeschränkten Testfall zeigt das neue Verfahren stabile Konvergenz.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF SYMBOLS

List of Symbols

Abbreviations

BFGS	Broyden-Fletcher-Goldfarb-Shanno algorithmm for Hessian updates
CPU	Central Processing Unit
FBE	Free Boundary Equilibrium
FE	Finite Element
FONC	First Order Necessary Conditions
ITER	International Thermonuclear Experimental Reactor: a tokamak reactor under construction in Cadarache, France
JET	Joint European Torus: a tokamak located in Culham, UK
KKT	Karush-Kuhn-Tucker (conditions), FONC for optimality
LCFS	Last Closed Flux Surface
PDE	Partial Differential Equation
PFC	Plasma-Facing Component
PM	Perturbation Model (for poloidal magnetic flux)
PM	Perturbation Model
prSQP	partially reduced Sequential Quadratic Programming
rSQP	reduced Sequential Quadratic Programming
SOA	Second Order Adjoint
SOL	Scrape-Off Layer
SQP	Sequential Quadratic Programming
WEST	tungsten (W) Environment in Steady-state Tokamak: a tokamak in Cadarache, France

Constants

$\varepsilon_{\mathbf{m}}$	Machine precision ($2.2 \cdot 10^{-16}$)
μ_0	Magnetic permeability of vacuum ($4\pi 10^{-7}\,\mathrm{H\,m^{-1}}$)

Geometry

h_θ	Metric coefficient, poloidal direction
h_r	Metric coefficient, radial direction
$\boldsymbol{\nu}$	Outwards-directed unit normal
$\sqrt{g}$	Jacobian of metric transformation
$\boldsymbol{e}_{(\cdot)}$	Unit vector in the $(\cdot)$ direction
R	Radial coordinate in a cilindrical coordinate system
Z	Height coordinate in a cilindrical coordinate system

Magnetic field

$\boldsymbol{A}$	Magnetic vector potential [Tm]
$\boldsymbol{B}$	Magnetic field vector [T]
$F(\psi)$	Flux function of R-scaled toroidal magnetic field
I_{P}	Total toroidal plasma current [A]

$\boldsymbol{J}$	Electric current density vector $\left[\mathrm{Am}^{-2}\right]$
ψ	Poloidal magnetic flux per radian $\left[\mathrm{Wbrad}^{-1}\right]$
$\bar{\psi}$	Normalized poloidal magnetic flux $[-]$
μ	Magnetic permeability
$p(\psi)$	Pressure flux function
b_θ	Magnetic field pitch $[-]$

Optimization

$\mathbb{A}$	Index set of active constraints at optimum
β_{max}	Maximal curvature bound
β_{min}	Minimal curvature bound
$\boldsymbol{c}$	State equations
$\widetilde{\boldsymbol{c}}$	Partially reduced state equations
$\boldsymbol{\varphi}$	Design variables
$\boldsymbol{\varphi}^\star$	Projected design variables
$\boldsymbol{G}$	State equations in fixed-point iteration form
$\boldsymbol{h}$	Design constraints
$\mathcal{I}$	Objective/cost functional
$\mathbb{I}$	Index set of inactive constraints at optimum
$\widetilde{\mathcal{I}}$	Partially reduced cost function
$\hat{\mathcal{I}}$	Reduced objective/cost functional
$\mathcal{L}$	Lagrangian
$\mathcal{L}_c$	Lagrangian including design constraints
λ_φ	Scaling variable for Tikhonov regularization $\left[\mathrm{A}^{-2}\right]$
λ_P	Scaling variable for wall heating penalty $[-]$
λ_Q	Scaling variable for heat load objective $\left[\mathrm{m}^2\mathrm{W}^{-2}\right]$
$\widetilde{\mathcal{L}}$	Partially reduced Lagrangian
L^a	Augmented Lagrangian
$\mathcal{F}$	State-dependent integrand of cost functional
$\boldsymbol{\mu}$	Lagrangian multipliers for inequality constraints
$\mathcal{N}$	Shifted Lagrangian
$\boldsymbol{q}$	State variables
$\boldsymbol{q}^*$	Adjoint variables or Lagrangian multipliers
$\boldsymbol{q}^*_{\mathcal{H}}$	Discretized continuous adjoint variables
$\boldsymbol{p}^*$	Adjoint boundary variables
ε	Step size
$\mathbf{y}^*$	Adjoint variables from fixed-point adjoint
A	Approximation to $\partial_{\boldsymbol{q}}\boldsymbol{c}$
B	Design preconditioner, Hessian estimate

Edge plasma

α_c	Core absorption coefficient $[-]$
α_p	Pump absorption coefficient $[-]$
$\mathbb{B}$	Big number
χ	Radial heat conductivity coefficient $\left[\mathrm{m}^2\mathrm{s}^{-1}\right]$
c_n	Neutral thermal speed $\left[\mathrm{ms}^{-1}\right]$

$\mathfrak{C}$	Boundary conditions
$\mathfrak{B}$	Plasma edge transport equations (domain)
c_s	Isothermal plasma sound speed $\left[\mathrm{ms}^{-1}\right]$
λ_n	Ion decay length [m]
λ_T	Temperature decay length [m]
D^i	Radial particle diffusion coefficient $\left[\mathrm{m}^2\mathrm{s}^{-1}\right]$
D_p^n	Neutral pressure diffusion coefficient $\left[\mathrm{m}^2\mathrm{s}^{-1}\mathrm{J}^{-1}\right]$
δ_sh	Sheath transmission coefficient $[-]$
E_i	Ionization energy [J]
E_p	Ionization potential [J]
η	Dynamic viscosity tensor $\left[\mathrm{kgm}^{-1}\mathrm{s}^{-1}\right]$
$\mathbf{\Gamma}$	Particle flux density $\left[\mathrm{m}^{-2}\mathrm{s}^{-1}\right]$
κ	Heat conductivity tensor $\left[\mathrm{m}^{-1}\mathrm{s}^{-1}\right]$
K_cx	Charge-exchange rate coefficient $\left[\mathrm{m}^3\mathrm{s}^{-1}\right]$
K_i	Ionization rate coefficient $\left[\mathrm{m}^3\mathrm{s}^{-1}\right]$
K_r	Recombination rate coefficient $\left[\mathrm{m}^3\mathrm{s}^{-1}\right]$
L_z	Radiative loss function $\left[\mathrm{Wm}^{-3}\right]$
m	Particle mass [kg]
n	Particle density $\left[\mathrm{m}^{-3}\right]$
L	Pumping speed $\left[\mathrm{m}^3\mathrm{s}^{-1}\right]$
ν^i	Radial kinematic viscosity of ions $\left[\mathrm{m}^2\mathrm{s}^{-1}\right]$
$\Omega_\mathbf{pe}$	Plasma edge simulation domain
p	Pressure [Pa]
$\boldsymbol{q}^*_\mathbf{pe}$	Plasma costate vector
Q_d	Desired energy flux density $\left[\mathrm{Wm}^{-2}\right]$
Q_o	Energy flux density to be optimized $\left[\mathrm{Wm}^{-2}\right]$
$\boldsymbol{q}_\mathbf{pe}$	Plasma state vector
$\mathbf{Q}$	Energy flux density $\left[\mathrm{Wm}^{-2}\right]$
R_c	Recycling coefficient $[-]$
Σ	Boundary of plasma edge simulation domain
T	Temperature [J]
C	Generalized convective flux or coefficient
D	Generalized conduction matrix or coefficient
u	Velocity component $\left[\mathrm{ms}^{-1}\right]$
$\mathbf{V}$	Velocity vector $\left[\mathrm{ms}^{-1}\right]$

Subscripts and superscripts

$'$	Fréchet-derivative
$^-$	Value at optimum
$\tilde{}$	Partially reduced function
$^\top$	Transposition
$_a$	Species a
$_G$	Function or parameter corresponding to Griewank's publications
$_h$	Discrete
$_r$	Radial direction

$_{\mathrm{e}}$	Electrons
$_{\mathrm{i}}$	Ions
$_{\mathrm{n}}$	Neutrals
$_{\perp}$	Diamagnetic direction
$_{\phi}$	Toroidal direction
$_{\boldsymbol{q}}$	Differentiation with respect to state
$_{\mathbf{eq}}$	Magnetic equilibrium
$_{\mathbf{gg}}$	Grid generation
$_{\mathbf{pe}}$	Plasma edge transport
$_{\theta}$	Poloidal direction
*	Adjoint complement

Operators

$\{\cdot\}_{\mathcal{H}}$	Discretization operator
Δ^{*}	Second order differential operator defined in equation (3.9)
δ	Differential, directional derivative
$\langle \cdot, \cdot \rangle$	Inner product
$\Delta^{\varepsilon}_{(\cdot)}$	Finite difference approximation of directional (partial) derivative (with respect to optional subscript), with step size ε
$\mathcal{P}$	Projection onto feasible set (see (4.7))
$\nabla_{(\cdot)}$	Gradient operator (with respect to optional subscript)
$\partial_{(\cdot)}$	Partial derivative with respect to subscripted variable
L	Second order differential operator defined in equation (3.13)

Other symbols

$\mathbb{1}$	Indicator function

1

Introduction

With an ever growing demand for energy and increasing environmental concerns, mankind faces the challenge to conceive a sustainable energy supply. In this process, all viable options for long term energy supply, compatible with our environment, will need thorough exploration. Amongst these, fusion energy stands out as a prominent option for reliable and safe energy supply not in the least because of the carbon-free, easily accessible, and virtually inexhaustible fuel. With the tokamak building of ITER, the first reactor-scale fusion project, currently rising from the ground in Cadarache (France), fusion research faces the last hurdles on the road to a viable reactor design. Producing 500 MW of fusion energy, this machine will bridge between current-day experiments and tomorrow's fusion power plants. The design of an exhaust system for power and Helium ashes is one of the key issues to be resolved before such power plants can be built. This thesis aims to address this very same task by developing design tools that allow state-of-the-art physics models to be used in an innovative design approach. In this first introductory chapter, the objectives of this PhD thesis are situated within the broader perspective of nuclear fusion and computational engineering research.

1.1 Nuclear fusion: confining the energy of stars

In nuclear fusion reactions, light particles combine, thereby releasing a high amount of energy. In the universe, it is the main source of energy. All stars,

amongst which our sun, are driven by fusion reactions. Yet, the production of large amounts of fusion energy on earth remains a challenge. One of the conceptual challenges is producing enough energy to ignite man-made fusion reactors. The sun profits from a luxurious amount of space and an excellent gravitational confinement. At the same time, with an energy density as low as 280 W/m^3, it is clear that the amount of fusion reactions need to be significantly increased for any practical fusion-based power plant concept on earth.

To this end, the *cross sections* of fusion reactions are increased by exchanging the proton-proton reactions of the sun for a fuel based on *deuterium* (D) and *tritium* (T), two hydrogen (H) isotopes. Resources of deuterium are abundantly available as a stable isotope of hydrogen in water. Tritium on the other hand is a rare radioactive isotope of hydrogen, but can be bred in the fusion reactors themselves from Lithium, an element as common as lead on earth. Additionally, the pressure in a fusion reactor needs to be sufficiently high. In practice, this necessitates temperatures ten times higher than that of the sun's core ($\pm$150 million K). At these temperatures, atoms spontaneously dissociate to loose electrons and ions in a state of matter known as *plasma*. Finally, the energy of the particles has to be confined long enough for sufficient fusion reactions to take place. To this end, the highly energetic particles of the fusion plasma need a strong insulation from the cold solids of the containing vessel.

The most promising way to confine these plasma particles at present is using a *magnetic confinement*[†]. In presence of a strong magnetic field, the Lorentz force causes the charged plasma particles to gyrate around the magnetic field lines, bounding their so-called *guiding center* close to the magnetic field lines. In contrast, the motion along the magnetic field lines is virtually unrestricted. Two magnetically confined fusion concepts at present compete, the *tokamak* and the *stellarator*. Both concepts are illustrated in figure 1.1 . Because of its early experimental successes, the tokamak is at present the most mature of these concepts and the current favoured technique. In this concept, charged particles are

[†]Alternatively, so-called *inertial confinement* can be used to confine the plasma. In this concept, lasers compress the fuel so rapidly that the fuel does not have the time to expand and come in contact with the vessel. At current, the low efficiency of about 1 % of these lasers prevents demonstrating the feasibility of net power production [34].

effectively trapped in the center of a torus by a strong toroidal magnetic field. To avoid that the plasma accelerates to the outer vessel wall[†], an additional poloidal field component is created by inducing a toroidal plasma current with a central transformer. Combined, this results in a helical magnetic field as illustrated in figure 1.1. Because the tokamak concept relies on this transformer for the plasma current, it has to operate in a pulsed fashion[‡]. Nevertheless, operating scenarios are envisaged in which the pulses might be stretched to many hours [111].

In stellarators on the other hand, steady-state operation is inherent. This is achieved by a complex coil configuration that constitutes a twisted magnetic field with an intrinsically 3D layout. Additionally, the absence of strong plasma currents simplifies the control of the machine. However, due to their complex configuration, stellarators only gained momentum recently with the availability of powerful supercomputers to assist design. At the moment of writing, a large stellarator project called Wendelstein 7-X successfully entered operation. In the coming years, studies on this device might clarify whether stellarators have the potential to outperform tokamaks as future fusion devices.

The first reactor-size fusion machine ever built, ITER (International Thermonuclear Experimental Reactor, or 'the way' in Latin), will be a tokamak. Since bigger tokamaks have lower heat losses per fusion volume, energy confinement, and thus reactor efficiency, improves strongly with reactor size. Based on these scaling laws, the large ITER reactor is designed to produce 10 times more heat than it consumes for periods of more than 400 seconds. As such, it will bridge the gap between today's smaller experimental devices and tomorrow's demonstration power plants. Even if stellarators finally prove a better alternative, there is no doubt that the knowledge gained from experiments on the ITER facility will be highly needed in the design of a stellarator-based reactor concept. Indeed, many issues such as the methods for tritium breeding and the effect of high amounts of plasma heating by fusion reactions are similar. Meanwhile, design studies for

[†]A consequence of the $\boldsymbol{E}\times\boldsymbol{B}$ forces, indirectly arising due to the radial decay of the magnetic field.

[‡]Inducing a constant plasma current requires a linear increase of the transformer magnetic field, because of Faraday's induction law. As a consequence, the primary current in the inner transformer coils needs to increase continually during operation, necessitating pulsed operation.

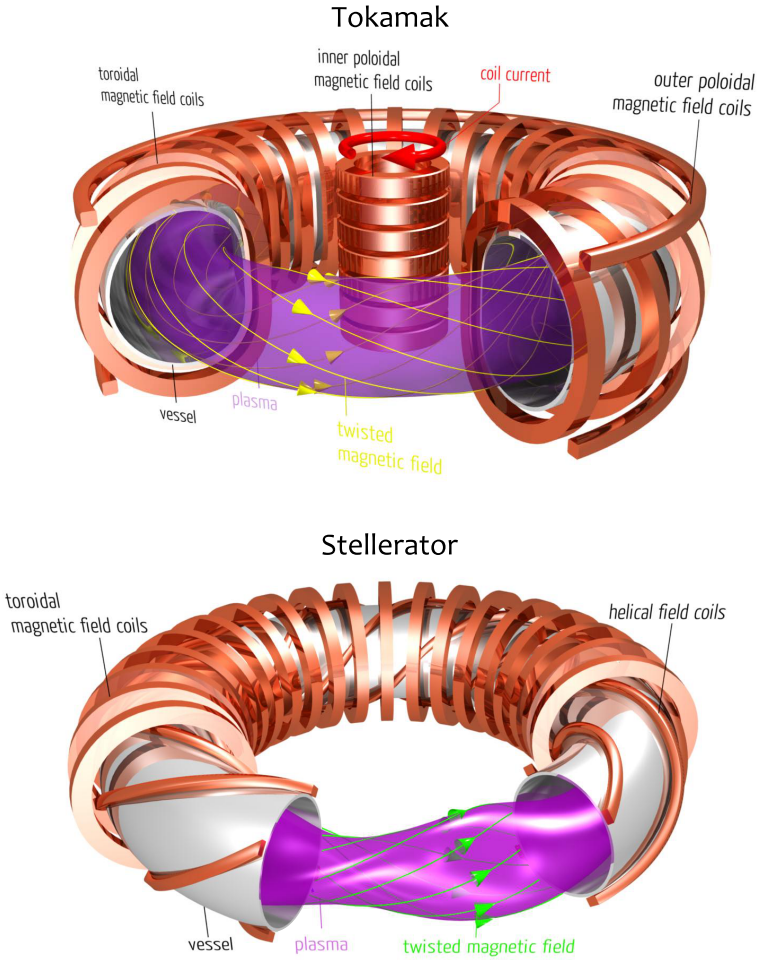


Figure 1.1: The tokamak and stellarator concepts: two magnetic confinement concepts for fusion reactors. The tokamak creates a helical and toroidally symmetric magnetic field using toroidal field coils and central transformer coils that induce a toroidal plasma current. In contrast, the complex 3D stellarator magnetic field does not need the central transformer. ©Max-Planck Institut für Plasmaphysik

the demonstration tokamak DEMO, that should prove the economical feasibility of fusion power, have already been initiated. Before DEMO can be constructed, a number of challenges need to be faced. These challenges are bundled in the EFDA roadmap [120]. One of them is the design of a tokamak particle and heat exhaust system that can deal with the heat fluxes of a commercial reactor. A topic that will be further elucidated in the next section.

1.2 The divertor: a tokamak exhaust system

In the previous section, the magnetic confinement of plasma particles in tokamaks was addressed. Although these charged particles have a strong tendency of following magnetic field lines, slow radial transport caused by turbulent advection and collisional processes will eventually transport particles to the edge of the reactor. As particles move further outwards, they will finally cross the so-called *last closed flux surface* (LCFS) that defines the last magnetic flux surface not intersecting the solid vessel. This LCFS can either be created by a protruding solid object called a *limiter* or by shaping the poloidal magnetic field using specific toroidal coils in order to create a saddle point of the poloidal magnetic flux, a so-called *X-point*, and *divert* the plasma fluxes towards a for the purpose designed structure, known as the *target plates* (see figure 1.2). This configuration is therefore called a *divertor* configuration and its LCFS is called the *separatrix*. The terminology of the divertor is illustrated in figure 1.3.

Because of the strong transport along the field lines, plasma particles crossing the LCFS will move very fast towards the solid limiter or divertor target surfaces. Since the slow radial processes do not get the chance to transport the particles much further outwards, the width of this so-called *scrape-off layer* (SOL) region is therefore not more than a couple of centimetres [133]. This SOL region between LCFS and vessel wall is the only region in the tokamak where the hot plasma meets cold vessel materials. It is characterized by a number of chemical interactions that govern complex transport processes. An understanding of these processes is essential to understand how particles reach the target and -above all- to what extent heat is deposited on the material surfaces. Understanding

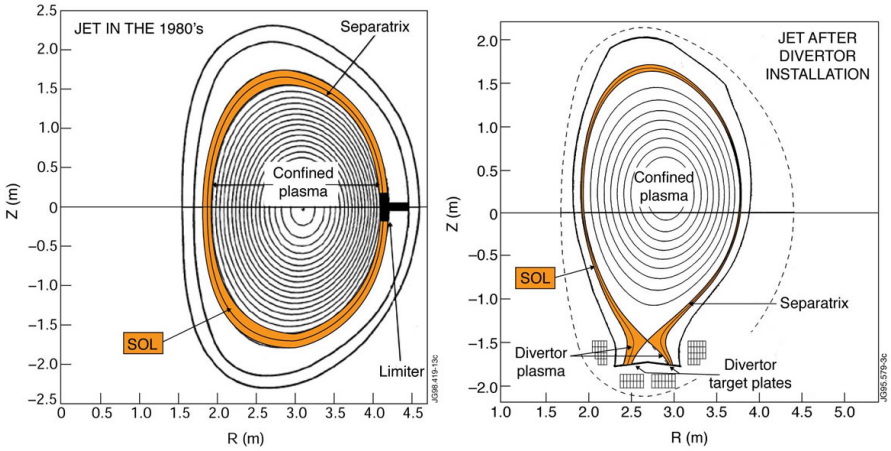


Figure 1.2: Illustration of the limiter and divertor scrape-off layer (SOL) in the Joint European Torus (JET) reactor. ©EFDA-JET

and modelling these *plasma surface interactions* is as such an important step in regard of the design of fusion reactors, discussed further in section 1.3.

1.2.1 Why today's tokamaks are equipped with divertors

Historically, first tokamaks were equipped with limiters. Yet, almost all of today's tokamaks feature a divertor configuration. For an extensive discussion on this choice the reader is referred to Refs. [53, 133]. Here, only a brief overview of the primary arguments is given.

Impurity transport

The original motivation to introduce divertor configurations was to reduce the impurity content in the main plasma. When plasma particles reach the cold vessel surface, electron and ions recombine to neutral atoms and molecules. These neutrals are no longer confined to the magnetic field. From the target, they are launched back into the plasma, where they can ionize again, a process called *recycling*. Simultaneously, when these plasma particles impinge the vessel, material of the vessel itself can be released due to *physical sputtering*, related to the impact

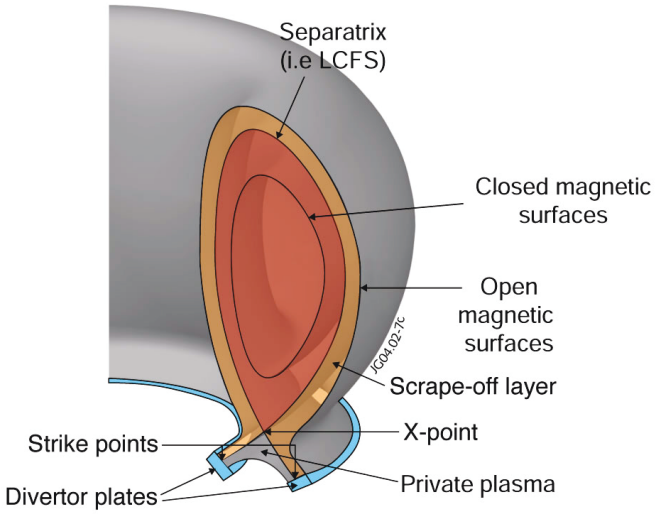


Figure 1.3: Schematic representation of the divertor configuration indicating terminology. ©EFDA-JET

of highly energetic particles and *chemical sputtering*, or erosion, due to chemical interactions between vessel and plasma[†]. These impurities are in general heavy atoms that tend to radiate heat from the plasma when ionized. If too much of these particles find their way to the core plasma, the increased radiation within the core has detrimental consequences on the energy confinement. In a divertor plasma, the majority of these impurities are born down at the divertor targets, far away from the main plasma. In the limiter case, on the contrary, they emerge close to the main plasma, resulting in much higher core impurity concentrations than in the divertor configuration.

Pumping

An important task in a nuclear fusion reactor is continuously removing these impurities and the Helium ashes, created in nuclear fusion reactions, from the

[†]Similar sputtering processes are present when highly energetic neutrals impinge vessel walls [133].

plasma. This is a necessity to avoid fuel dilution when operating on a longer time scale than the present experimental devices (a typical pulse is in the order of seconds). Only neutral particles can be efficiently pumped from the reactor. The divertor configuration allows rather naturally to compress the neutral particles near the pumping entrance. As such, compact pumping systems can be used. Additionally, the reactor control benefits from these good pumping capabilities since Hydrogen can be pumped to control the density in the reactor. In a pumped limiter configuration, the neutral pressures are much lower.

New plasma and divertor regimes

The most important reason for the dominance of the divertor SOL in present-day tokamak devices is the spontaneous occurrence of two highly beneficial phenomena: the high confinement regime (*H-mode*) and plasma *detachment*. The former is a phenomenon observed experimentally as the heating power is increased in a divertor tokamak discharge. At some point, turbulent transport in the edge plasma is drastically reduced, creating a so-called *transport barrier* [53]. At present, the H-mode regime is not well understood theoretically. Nevertheless, the enhanced energy confinement is of great advantage for the commercial exploitation of fusion reactors.

The earlier mentioned *recycling* process lies at the basis of the *detached* plasma regime. Plasma particles recombine to neutral atoms or molecules at the targets. If not pumped, these neutrals ionize again after collisions with plasma particles. Then, they augment the plasma flow that flows along the magnetic field towards the targets in a recycling process. When the density of the core plasma is increased, the recycling flux and the plasma-neutral collisions strongly increase. As such, more momentum is extracted from the plasma, effectively cooling the plasma before it reaches the target. When the density in the core is even further increased, the temperature at the targets is lowered eventually to temperatures below 5 eV at which volumetric recombination sets in. As such, a neutral cushion forms in front of the targets, actually *detaching* the plasma flow from the targets and protecting the integrity of the target tiles from excessive heat loads.

Power exhaust limitations

Unfortunately, the divertor configuration has some downsides too. A straightforward disadvantage is the need for installing additional coils to divert the magnetic field. In addition, the core region is smaller in divertor plasmas, which means that a smaller part of vacuum chamber can be used for fusion reactions. Both increase the cost of reactors with divertor plasmas considerably.

Moreover, one of the key problems in divertors SOL is that a relative large part of the heat is transported along with the edge plasma towards the targets and ultimately exhausted in a small zone on the target surface, the so-called *plasma wetted area*. Therefore, special heat-resistant materials will need to be used for the divertor targets. In ITER, tungsten tiles will be used that are designed to handle constant power loads of ~ 10 MW/m^2, twice as much as experienced on the nose cone of a space shuttle at re-entry into earth's atmosphere. Furthermore, the tokamak should be carefully operated to avoid transient energy loads exceeding ~ 0.5 MJ/m^2 in 250 μs [53]. Above these limits, excessive sputtering will destruct the target tiles. To meet the requirement on transient energy deposition, it is essential that strong periodic expulsions of plasma particles and heat from so-called *edge localized modes* (ELM's) are controlled. Also the steady-state material heat load limits have to be carefully considered. In the baseline ITER scenarios, the continuous target heat load will be reduced by:

1. Inclining the target surfaces with respect to the magnetic field to obtain a larger projected plasma wetted area. The potential of this method is limited to minimal glancing angles in order to prevent shadowing effects induced by small misalignment between the magnetic field and the tiles.

2. Puffing impurities in the divertor SOL to increase volumetric energy losses by radiation. Consequently, the radiated energy will be deposited more uniformly on the target surface, avoiding excessive heat peaks.

3. Triggering a (partially) detached plasma operating regime. This can be achieved by using a 'closed' divertor geometry, in which the neutrals are trapped in front of the target surface. The vertical target configuration in ITER will facilitate this [93].

However, as discussed in the next section, respecting materials limits will be a true challenge in commercial fusion reactors.

1.2.2 How to prevent excessive target erosion in DEMO?

Kotschenreuther et al. [85] estimated in a study based on simulations and scaling laws, that if the standard divertor is used in the commercial DEMO reactor, divertor target materials might be irreparably damaged. This is not surprising, since commercial fusion will require a maximal power production and thus high target heat loads. In comparison to ITER, the assumed fusion power output will be at least three times higher and the size will be increased with about 50% [51, 142]. That DEMO will be an even larger fusion reactor than ITER is mainly needed to ensure a sufficiently good energy confinement. Intuitively, this can be explained using a simple reasoning. Since in larger reactors the relative increase of fusion volume is proportionally larger than the increased area for heat losses, the energy confinement benefits from an increased reactor size. The target heat loading problem, in contrast, becomes worse because of exactly the same argument. **The EFDA roadmap [120] therefore lists the development of an adequate solution for the high divertor power exhaust in DEMO as one of the key challenges for confined fusion research.**

The design of the divertor configuration is thus a crucial exercise with often conflicting requirements: maintaining the heat load on the divertor targets steadily below 10 $\mathrm{MW/m^2}$, the heat exhaust limit of state-of-the-art materials, while maintaining good pumping capabilities[†]. In order to meet these requirements in DEMO, large amounts (over 50%) of the energy entering the SOL will have to be disposed of through impurity radiation. Furthermore, current fusion research actively examines alternative magnetic divertor configurations that might comply better to these requirements [50, 142]. As such, novel divertor configurations with increased plasma-wetted area have recently been proposed:

[†]A parallel research track aims at developing advanced plasma-facing components that could exhaust still higher heat loads. In this context, liquid metal targets are amongst others examined [50, 142]. Yet, plasma contamination by the liquid metal and influence of MHD forces on the liquid targets pose some serious challenges for these concepts.

the 'X-divertor' [85], the 'Super X-divertor' [143], and the 'Snowflake divertor' [122, 123, 124].

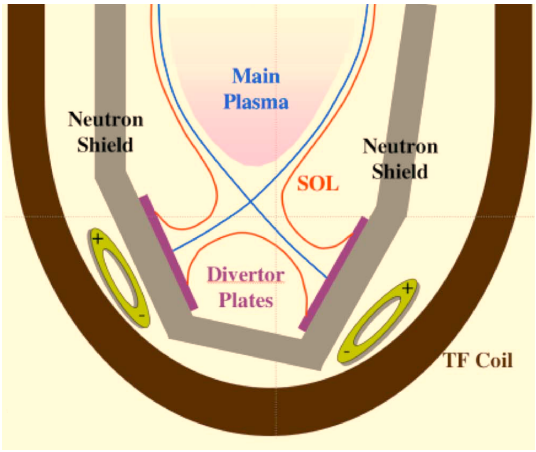


Figure 1.4: Illustration of the X-divertor configuration. At each side of the reactor, additional coils are introduced to create a local expansion of poloidal magnetic flux. As such, the concept is a proposal to increase the plasma-wetted area and lower target heat loads. Figure reproduced from Ref. [85].

These concepts are often based on rather simple (geometric) reasoning. In the X-divertor configuration, additional coils are used to create a local flux expansion towards the divertor targets (see figure 1.4). As such, the plasma is smeared out over the target surface, effectively increasing the plasma-wetted area and reducing the heat load. The super X-divertor concept additionally uses very long divertor legs. Consequently, the distance of particles travelling from the core plasma to target surfaces, the so-called *connection length*, is increased and plasma has more time to cool down by volumetric processes or by radial heat transport. Furthermore, the outer divertor leg is located entirely at outer radius of the torus to profit from additional flux expansion in the toroidal direction. The snowflake configuration, finally, uses a second order null of the magnetic flux to exhibit a snowflake-like divertor shape (see figure 1.5). Due to the second-order null, an extreme flux expansion is achieved.

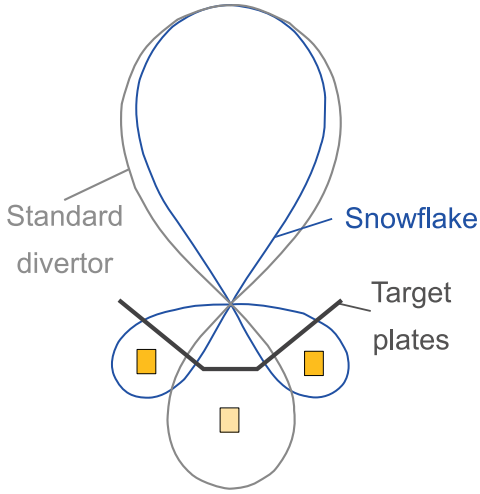


Figure 1.5: Schematic illustration of the difference between the separatrix in an exact snowflake (blue) and a standard divertor configuration (grey). The second order null in magnetic flux creates a snowflake-resembling magnetic divertor shape.

These concepts clearly indicate the potential of magnetic divertor design to provide a solution for the heat exhaust. Further studies on these alternative configurations and their characteristic plasma physics are a priority in current fusion research [50, 120].

1.3 Computational divertor design

In absence of experiments with comparable parameters, design of ITER and DEMO requires extrapolating current knowledge of edge plasma physics to these reactor-sized machines. While scaling laws can be used to estimate the plasma conditions in these reactors, a more reliable extrapolation can be achieved using physics based models. *Plasma edge models* can as such be used to identify operational windows for plasma detachment and maximal fractions of SOL radiation in ITER and DEMO. Such models furthermore play a key role in the interpretation of experiment and understanding the edge plasma physics.

Plasma edge codes model the transport processes in the SOL. These codes feature a multifluid Navier-Stokes-like description of the plasma edge transport, with equations for particle flux, momentum, and energy conservation, and are typically discretized using finite volume methods. The highly kinetic neutral atoms and molecules, however, need a different approach. For these neutrals, Monte Carlo codes are used to statistically approximate the solution of the kinetic equations. This leads to a set of partial differential equations that is strongly coupled through source terms to account for the many collisional processes.

Although the plasma transport is in principle described by a form of the *magnetohydrodynamics* (MHD) equations, this approach is too difficult for practical simulations of edge flow. Instead, the low current density in the edge plasma justifies approximating the macroscopic MHD equilibrium as independent of the edge flow. As such, the plasma edge simulations are performed using a fixed magnetic field that is obtained from an (approximate) MHD solution[†]. The main task of the plasma edge simulation then reduces to calculating the plasma transport and interactions in the SOL, given imposed plasma conditions at a chosen core-boundary interface. Additionally, the smallest time scales in edge codes are averaged out, to come to a mean-flow description based on the so-called *Braginskii equations* [25] [‡]. Given the strong coupling between plasma and neutral equations and the many impurity species with different ionized states (tungsten has 74 of them!), plasma edge simulations still remain highly challenging.

In conclusion, edge plasma equations are solved using a hybrid fluid-kinetic approach that combines a *finite volume method* for the plasma particles with a Monte Carlo method for the neutrals. The fluid equations are typically discretized on quadrilateral curvilinear meshes. To accommodate for the strong flow anisotropy and to circumvent excessive numerical diffusion, these meshes are aligned to the poloidal magnetic flux surfaces. Examples of such hybrid codes are B2-EIRENE [115], EDGE2D-NIMBUS [130]. EDGE2D-EIRENE [150], SOLEDGE2D-EIRENE [27], and UEDGE-DEGAS [118, 135]. Especially the B2-EIRENE code is used worldwide for analysis of experimental tokamaks [29, 84].

[†]In practice, the evaluation of the magnetic equilibrium is often based on the solution of 1.5D Grad-Shafranov equation for ideal static MHD equilibrium [60, 128].

[‡]In some codes, slightly different closure schemes are used according to Balescu [6].

Moreover, the SOLPS code package that includes the B2-EIRENE code has been the workhorse for the design of the ITER divertor [86, 87, 88, 89] and to study possible operation scenarios with impurity seeding [108]. Recently, it was adopted by ITER as the standard tool for ITER divertor modelling in a new SOLPS-ITER version [151]. Also for DEMO, B2-EIRENE simulations play an important role to extrapolate current knowledge to reactor-scale divertor design [107].

Nevertheless, the computational time of the B2-EIRENE code is a serious bottleneck in its use for design applications. In fact, CPU-time of a single simulation might range up to a year (paralleled over a cluster) for simulation of power plant relevant conditions [41]. Add to this the large number of design variables (coil configuration, plate configuration, and plasma state controls), and one immediately realizes that divertor design is extremely CPU-demanding. Nevertheless, present divertor design is based on a multitude of rather time-consuming one-dimensional parameter scans. These studies therefore preclude by definition the investigation of a wide range of design choices and operational points. Given the need for better performing configurations, **divertor design studies could greatly benefit from the use of more efficient design approaches**.

1.4 Optimal design

In other disciplines, similar design problems have been tackled using *optimal design* methods. Introduced by Lions in the 70's [92], optimal design strategies for problems governed by *partial differential equations* (PDEs) have meanwhile shown their virtues in many design applications, including a range of problems in i.a. aerodynamics, structural mechanics, inverse imaging, chemical sciences, heat transfer, and quantum mechanics (see e.g. Refs. [21, 109, 138] for several industrial applications). For instance, a popular aerodynamics application is the optimal drag reduction by shape design of a wing profile governed by the Navier-Stokes equations [80]. The design goal is then recast into a so-called *cost function* (the total drag force on the wing), which is minimized as a function of the *design variables* (the parametrization of the wing profile).

Because of the complex and large-scale nature of these PDEs, specific algorithms are needed to efficiently solve these optimization problems. For a thorough

discussion of this subject, the reader is referred to Refs. [21, 73, 141]. Typically, *adjoint* based optimization methods are used to reduce computational cost. In fluid dynamics applications, they were first introduced by Pironneau [110]. After solving the model or *state equations*, an additional set of *adjoint equations* is then solved to find the sensitivity of the cost function with respect to the design variables. These adjoint equations can be derived both for the continuous optimization problem or for the discretized optimization problem, known as the *continuous adjoint* or *discrete adjoint* approach, respectively [59]. Using the adjoint method, the complete sensitivity vector can as such be calculated at the computational cost of roughly two state simulations. This is in strong contrast to *direct* sensitivity calculations, for which the computational cost scales linearly with the number of design variables. Because of this property, the adjoint method enables design applications with a very high number of design variables, such as shape or topological optimization. To find the optimum of the cost function, these adjoint sensitivities are then used in a gradient based optimization algorithm.

The convergence of this so-called *nested* optimization strategy that performs a complete iterative solution of model and adjoint equations in each optimization step can be significantly accelerated by the use of *one-shot* methods [90, 136]. These methods simultaneously converge the model equations, adjoint equations, and the *design equation*, that represents the final condition for optimality. Different variants of the one-shot method have been successfully applied in aerodynamics design [22, 54, 68, 69, 90, 105] and inverse modelling applications [22, 78]. One of the key features of one-shot methods is the non-intrusive way in which the PDE-solver is incorporated in the optimization approach. Using one-shot optimization, a typical equivalent computational cost of roughly 5-10 flow simulations is often found to suffice for the entire optimization process. However, guaranteeing convergence of one-shot methods has long been a difficulty. State-of-the-art one-shot methods now feature embedded line search strategies based on exact penalty functions to overcome this issue and obtain global convergence [65, 66].

In nuclear fusion applications, adjoint based sensitivity calculations have been used in the reconstruction of magnetic equilibria [74, 91, 98] and for optimal control of the magnetic equilibrium [18, 20, 72]. Recently, Dekeyser successfully

applied one-shot methods for optimal design of the divertor target shape in his PhD thesis [37, 38, 39, 40, 41, 42, 43]. In a first step, a slightly reduced version of the B2-EIRENE code with a fluid neutral model was used to model the transport in the edge plasma. Using a continuous adjoint framework and a non-parametric approach for shape sensitivities, nearly perfectly uniform target heat loads were achieved using a tracking-type objective functional for the heat load [39]. The resulting strongly inclined and V-shaped targets are features that are retained in the ITER divertor [89] and show the potential of the method to pinpoint beneficial operating conditions. Moreover, the one-shot method enabled a full optimal design within a computational time of 10 equivalent single runs. The model has finally been extended with an adjoint Monte-Carlo method to account for the effect of non-uniform radiation on the target surface.

The promising results of Dekeyser lie at the basis of this thesis work. **Here, adjoint-based optimal design methods will now be pursued for design of the magnetic divertor configuration**. Based on the recent striving within the fusion community towards improved magnetic configurations, enabling automated design of the magnetic configuration is expected to even further increase the power of the novel design tool. Used in combination with divertor shape design, these tools are able to assist fusion research in finding a commercially viable operating window in DEMO. Besides, in contrast to design of the target shape, such a tool for magnetic divertor design might also find its use to set up experiments in existing reactors, since no reactor components need replacement.

1.5 Goals and outline of the thesis

The primary objective of the thesis is the selection and development of suitable models and optimal design strategies that meet the specific needs of magnetic divertor design in nuclear fusion reactors. The aim is to come to an automated design strategy that is able to retrieve magnetic divertor configurations with improved capabilities to tackle the tremendous heat loads on the target plates. Moreover, these design approaches need to account for the relevant design constraints.

The methods are developed for eventual application on the full B2-EIRENE code suite. This imposes several additional requirements on the optimization strategies. First of all, the highest possible efficiency is needed to accommodate for the high CPU-cost of the complex simulations. Secondly, the ease of implementation will be of primary importance. And finally, robustness will be a key element in the implementation of the automated design strategy.

Since optimal design methods are state-of-the-art in the fusion community and only recently pioneered by Dekeyser [41], the added value and use of the optimal design strategy is extensively illustrated. Throughout the text, realistic examples for current experimental tokamak devices are given that demonstrate the possibilities of the methodology. Furthermore, the concepts and terminology of the optimization community will be carefully introduced. Therefore, this monograph is initiated with an introductory chapter on relevant optimization concepts.

In present-day tokamak analysis, the calculations of magnetic equilibrium and plasma edge transport calculations are treated separately. The first accomplishment of this thesis is therefore to extend the plasma edge model of Dekeyser [41] with a model for magnetic field calculations. Initially, a perturbation model is set up for this purpose. Furthermore, since the computational mesh needs to be aligned to the magnetic equilibrium for accurate transport calculations, the grid generator becomes an integral part of this integrated code. To achieve the first integrated model that is able to evaluate the influence of divertor coils on the target heat load in an automated fashion, an autonomous grid generator is therefore developed. These subtle considerations with respect to modelling are treated extensively in chapter 3.

In 4, the first automated design of a magnetic divertor configuration is presented. To this end, cost function and design constraints are carefully set up to avoid that the optimization routine would exploit any modelling flaws. An inequality-constrained nested quasi-Newton optimization routine is then implemented to solve the PDE-constrained design problem. For robustness, a novel globalization strategy is developed that combines an efficient line search algorithm based on the generalized Wolfe conditions with a nonlinear gradient projection to guarantee design feasibility. Results are presented and interpreted for a real-

istic design study on the Joint European Torus (JET), the largest experimental tokamak currently in operation.

Where the first magnetic divertor optimization study is based on finite difference evaluations of the gradient, the power of adjoint based methods is used in chapter 5 to accelerate the design process. Given the complexity of the integrated code with dependencies of the plasma edge domain on the magnetic equilibrium, a full adjoint sensitivity calculation (discrete or continuous) is judged difficult and impractical. A practical procedure to sensitivity calculation, which is in-parts discrete finite differences, in-parts continuous adjoint, is therefore derived and implemented. Special attention is hereby paid to the consistency between the implemented boundary conditions of the plasma edge transport model and the boundary multipliers from its adjoint counterpart. Using this method, a similar acceleration of the sensitivity calculations is achieved as that for a full continuous adjoint method. Furthermore, by exploiting forward sensitivity calculations on the inexpensive part of the model, sensitivities to virtually any chosen code parameter can be calculated with this method without the usual derivational effort of the continuous adjoint procedure. The sensitivities are verified with a grid refinement study and used to treat a more challenging magnetic divertor design problem, with an increased number of design variables.

In chapter 6, the magnetic equilibrium model, one of the suspected weaknesses in the model, is revisited. For this purpose, the model discrepancies with the CEDRES++ finite element code for free boundary static MHD equilibrium calculations are examined. Based on the results of this study, the CEDRES++ code is entirely integrated into the magnetic divertor design tool to improve the accuracy of magnetic field and sensitivity evaluations. Again, in-parts adjoint sensitivity calculations are used for efficient calculation of sensitivities. The extended optimal design framework is then demonstrated on the WEST divertor, a new divertor currently under construction in the experimental tokamak at CEA (Commissariat á l'Energie Atomique et aux Energies Alternatives), France.

In chapter 7, one-shot methods are considered for a further acceleration of the automated magnetic divertor design. The chapter starts with a literature review of one-shot methods. First, a basic one-shot method is then applied based on the

simultaneous pseudo-time stepping method of Hazra et al. [69]. Next, a grid deformation method is derived for field-aligned curvilinear grids to reduce the cost of design steps. Afterwards, one-shot globalization theory from literature [66] is adapted to come to a novel globalized one-shot method that is more efficient and better suited for practical applications. The novel method features a line search strategy based on an augmented Lagrangian penalty function that systematically accounts for the feasibility of state and adjoint equations. Furthermore, a consistent version of the well-known BFGS (Broyden-Fletcher-Goldfarb-Shanno) approach is derived for superlinear convergence of one-shot optimization methods, along with a practical strategy to bound the Hessian. Reasonable adaptations are then made to come to a practical implementation based on the in-parts continuous adjoint sensitivities. First results are shown for an unconstrained magnetic divertor design problem. Extensions to a constrained version of the one-shot method are discussed in an outlook section.

In chapter 8, the thesis is closed with a resume of the general conclusions and an outlook to a number of promising tracks for future research.

2

Optimal Magnetic Divertor Design

The objective of this PhD is to examine the suitability of adjoint-based optimal design methods for magnetic divertor design. Although these methods have been well established for aeronautics design, these methods have barely been explored in nuclear fusion design applications. Therefore, concepts, techniques, and notations relevant for this particular optimization problem are introduced in this chapter. First, the mathematical formulation of the optimization problem, the optimality conditions and adjoint-based sensitivity calculation are introduced in section 2.1. Next, in section 2.2, some known solution methods for optimization problems governed by partial differential equations are summarized. Finally, globalization strategies, which avoid divergence of the iterative optimization procedures, are briefly introduced in section 2.3. For further notions on general optimization methods, the reader is referred to Ref. [103]. A book-length treatment of the specific application to PDE-constrained optimization problems can be found in Refs. [21, 73, 141].

2.1 Divertor design as an optimization problem

This thesis tackles the design problem of finding the divertor coil configuration that leads to the best possible heat load at the divertor targets. The first step

to obtain an optimal design consists of reformulating the design problem into a mathematical optimization problem with constraints:

$$\begin{aligned} \min_{\boldsymbol{\varphi}\in\Phi_{\text{ad}},\boldsymbol{q}} \quad & \mathcal{J}\left(\boldsymbol{\varphi},\boldsymbol{q}\right) \\ s.t. \quad & \boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q}) = 0, \end{aligned} \tag{2.1}$$

where the distinction is made between two types of variables. The first are the *control* or *design variables* $\boldsymbol{\varphi} \in X$, which are the independent variables that need to be optimized. In this thesis, the controls of interest are clearly related to the divertor coil currents that govern the magnetic divertor configuration. The second are the *state variables* $\boldsymbol{q} \in Y$, which are dependent variables that can be uniquely defined from the control variables by the *state* or *model equations* $\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q}) = 0$, $\boldsymbol{c} : X \times Y \mapsto Y$. The state variables can therefore also be written in short as a function $\boldsymbol{q}(\boldsymbol{\varphi})$ of the design variables. We assume that X and Y are Hilbert spaces. This means that they are complete spaces equipped with an inner product $\langle\cdot,\cdot\rangle$.

Particular for this optimization problem is that the model equations under consideration are a set of partial differential equations (PDEs) that can only be solved numerically at high computational cost using spatial discretization methods. The models that will be used to evaluate the target heat load are considered in chapters 3 and 6, but they could for example include the Braginskii equations [25] for plasma edge transport, solved for state variables $\boldsymbol{q}$ such as the density of deuterium. Often, these PDEs $\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q}) = 0$ are solved in their weak form. A set of test functions $\boldsymbol{q}^*$ is hereby introduced to obtain the equivalent variational form of the equations,

$$\langle \boldsymbol{q}^*, \boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})\rangle = 0 \qquad \forall \boldsymbol{q}^* \in Y. \tag{2.2}$$

Remark that throughout this thesis, a suitable inner product choice is assumed that accounts for the dimensionality of the arguments. Consider $\boldsymbol{c}_\Omega = 0$ as the set of PDEs defined on a domain Ω, with boundary conditions $\boldsymbol{c}_\Sigma = 0$ on the domain boundary $\Sigma \equiv \partial\Omega$, such that $\boldsymbol{c} = [\boldsymbol{c}_\Omega, \boldsymbol{c}_\Sigma]^\top$. One could then choose to use the standard L^2 inner product for domain contribution as well as boundary

conditions to obtain

$$\langle \boldsymbol{q}^*, \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) \rangle = \int_\Omega \boldsymbol{q}^*_\Omega \cdot \boldsymbol{c}_\Omega d\omega + \int_\Sigma \boldsymbol{q}^*_\Sigma \cdot \boldsymbol{c}_\Sigma d\sigma, \tag{2.3}$$

with test functions $\boldsymbol{q}^* = [\boldsymbol{q}^*_\Omega, \boldsymbol{q}^*_\Sigma]^\top$.

The scalar-valued objective $\mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{q}) : X \times Y \mapsto \mathbb{R}$ is the mathematical representation of the design goal. Remark that the state equations $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) = 0$ can be used to eliminate the state variables $\boldsymbol{q}(\boldsymbol{\varphi})$ from the optimization problem (2.1). We can thus define the reduced objective $\hat{\mathcal{I}}(\boldsymbol{\varphi}) := \mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{q}(\boldsymbol{\varphi}))$. We assume that the feasible or admissible set of control variables $\Phi_{\text{ad}} \subset X$ is a convex subset of the Hilbert space X. The feasible set can be represented by $\Phi_{\text{ad}} = \{\boldsymbol{\varphi} \in X | \ \boldsymbol{h}(\boldsymbol{\varphi}) \leqslant 0, \boldsymbol{h} : X \mapsto H\}$, where $\boldsymbol{h}(\boldsymbol{\varphi}) \leqslant 0$ are so-called design constraints and H is a Hilbert space as well. The choice of the design objective $\mathcal{I}$ and constraints $\boldsymbol{h}$ is treated in chapter 4.

2.1.1 First order necessary conditions for optimality

Before introducing solution procedures to solve the optimization problem (2.1), the criteria are listed that indicate when $\boldsymbol{\varphi}$ and $\boldsymbol{q}$ reach an optimal value $\bar{\boldsymbol{\varphi}}$ and $\bar{\boldsymbol{q}}$, respectively. It will be conveniently assumed that $\mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{q})$, $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q})$ and $\boldsymbol{h}(\boldsymbol{\varphi})$ are continuously Fréchet-differentiable with respect to their arguments. For the objective, this means that for all directions $\boldsymbol{\delta\varphi}$, it is possible to evaluate the directional derivative $\delta\hat{\mathcal{I}}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi}) = \hat{\mathcal{I}}'(\boldsymbol{\varphi})\boldsymbol{\delta\varphi}$, where the Fréchet-derivative is denoted using a prime. The gradient operator ∇ can then be defined from the Riesz-representation of the directional derivative,

$$\delta\hat{\mathcal{I}}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi}) \equiv \left\langle \nabla\hat{\mathcal{I}}(\boldsymbol{\varphi}), \boldsymbol{\delta\varphi} \right\rangle \qquad \forall \boldsymbol{\delta\varphi} \in X. \tag{2.4}$$

In the specific case that the Hilbert space of the design variables, X, is finite-dimensional, the relation between gradient and derivative is given by

$$\nabla\hat{\mathcal{I}}(\boldsymbol{\varphi}) = \hat{\mathcal{I}}'(\boldsymbol{\varphi})^\top, \tag{2.5}$$

where $(\cdot)^\top$ stands for transposition. A partial (Fréchet-)differentiation $\partial\cdot/\partial a$ with respect to a variable 'a' is indicated further using the shorthand notation $\partial_a(\cdot)$.

Similarly, a subscripted ∇-operator will denote the gradient only with respect to the variable in subscript. E.g., the gradient of the cost functional with respect to the design variables can then be related to the partial derivative $\partial_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q})$ through

$$\partial_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q})\boldsymbol{\delta\varphi} \equiv \left\langle \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q}), \boldsymbol{\delta\varphi}\right\rangle \qquad \forall \boldsymbol{\delta\varphi} \in X. \tag{2.6}$$

The Karush-Kuhn-Tucker (KKT) first order necessary conditions for optimality of the state-constrained optimization problem are then found by equating the derivatives of the Lagrangian

$$\mathcal{L}(\boldsymbol{\varphi},\boldsymbol{q},\boldsymbol{q}^*) = \mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q}) + \left\langle \boldsymbol{q}^*, \boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})\right\rangle \tag{2.7}$$

to zero, with $\boldsymbol{q}^*$ the so-called Lagrange multipliers. The KKT conditions for an optimal point $(\bar{\boldsymbol{\varphi}},\bar{\boldsymbol{q}},\bar{\boldsymbol{q}}^*)$ thus read

$$\begin{cases} \nabla_{\boldsymbol{q}^*}\mathcal{L} = \boldsymbol{c}\left(\bar{\boldsymbol{\varphi}},\bar{\boldsymbol{q}}\right) = 0 \\ \nabla_{\boldsymbol{q}}\mathcal{L} = \nabla_{\boldsymbol{q}}\mathcal{J}(\bar{\boldsymbol{\varphi}},\bar{\boldsymbol{q}}) + \partial_{\boldsymbol{q}}\boldsymbol{c}^*(\bar{\boldsymbol{\varphi}},\bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^* = 0 \\ \left\langle \nabla_{\boldsymbol{\varphi}}\mathcal{L}, \boldsymbol{\delta\varphi}\right\rangle = \left\langle \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\bar{\boldsymbol{\varphi}},\bar{\boldsymbol{q}}) + \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^*(\bar{\boldsymbol{\varphi}},\bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^*, \boldsymbol{\delta\varphi}\right\rangle \geqslant 0 \quad \forall \boldsymbol{\delta\varphi} \in \Psi_{\text{ad}} \end{cases}, \tag{2.8}$$

with Ψ_{ad} the set of feasible design perturbations corresponding to the feasible set of design variables as $\Psi_{\text{ad}} = \Phi_{\text{ad}} - \bar{\boldsymbol{\varphi}}$. The second and third sets of equations in these KKT conditions include the adjoint operators $\partial_{\boldsymbol{q}}\boldsymbol{c}^*$ and $\partial_{\boldsymbol{\varphi}}\boldsymbol{c}^*$, respectively. In general, the adjoint operator A^* of a bounded linear operator A is defined with respect to the inner product $\langle\cdot,\cdot\rangle$ through

$$\langle x, Ay\rangle = \langle A^*x, y\rangle\,. \tag{2.9}$$

One can easily observe that the first set of equations in these KKT conditions corresponds to the state equations. The second set of equations is called the *adjoint equations* and the Lagrange multipliers $\boldsymbol{q}^*$ that solve them are *adjoint* or *dual variables*. This set of equations is found by differentiating the Lagrangian with respect to the state variables $\boldsymbol{q}$ and then using the adjoint identity

$$\left\langle \boldsymbol{q}^*, \partial_{\boldsymbol{q}}\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})\boldsymbol{\delta q}\right\rangle = \left\langle \partial_{\boldsymbol{q}}\boldsymbol{c}^*(\boldsymbol{\varphi},\boldsymbol{q})\boldsymbol{q}^*, \boldsymbol{\delta q}\right\rangle. \tag{2.10}$$

The last set of equations in (2.8) are called the design equations. When state and adjoint equations are feasible and when no design constraints are applied, this

simply reduces to $\nabla\hat{\mathcal{J}} = 0$ (see section 2.1.2). As indicated in figure 2.1, these KKT-conditions might hold multiple solutions $\bar{\boldsymbol{\varphi}}$, including maxima or saddle points in a higher dimensional design space. To have a minimum, it is then sufficient that the Hessian of $\hat{\mathcal{J}}$ at $\bar{\boldsymbol{\varphi}}$ is positive definite.

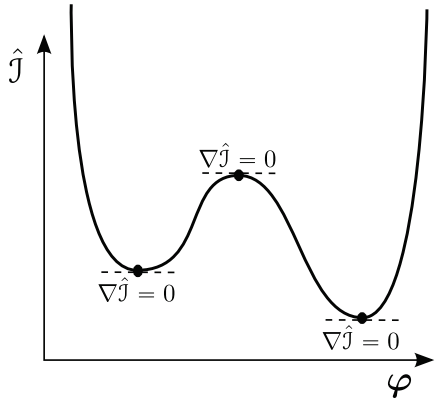


Figure 2.1: A one-dimensional cost functional landscape that demonstrates how the KKT-conditions might yield multiple solutions. Notice that these solutions might include maxima as illustrated here.

Whereas in equation (2.8) the inequality constraints $\boldsymbol{h}(\boldsymbol{\varphi}) \leqslant 0$ are taken into account implicitly in the feasible set Ψ_{ad}, it is also possible to explicitly account for them in the KKT conditions. Starting from the Lagrangian for the design constrained problem

$$\mathcal{L}_c(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*, \boldsymbol{\mu}) = \mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}) + \langle \boldsymbol{q}^*, \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) \rangle + \langle \boldsymbol{\mu}, \boldsymbol{h}(\boldsymbol{\varphi}) \rangle, \tag{2.11}$$

with Lagrange multipliers $\boldsymbol{\mu} \in H$, the full KKT system yields

$$\begin{cases} \nabla_{\boldsymbol{q}^*}\mathcal{L}_c = \boldsymbol{c}\,(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}) = 0 \\ \nabla_{\boldsymbol{q}}\mathcal{L}_c = \nabla_{\boldsymbol{q}}\mathcal{J}(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}) + \partial_{\boldsymbol{q}}\boldsymbol{c}^*(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^* = 0 \\ \nabla_{\boldsymbol{\varphi}}\mathcal{L}_c = \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}) + \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^*(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^* + \partial_{\boldsymbol{\varphi}}\boldsymbol{h}^*(\bar{\boldsymbol{\varphi}})\bar{\boldsymbol{\mu}} = 0 \\ \boldsymbol{h}(\bar{\boldsymbol{\varphi}}) \leqslant 0 \\ \bar{\boldsymbol{\mu}} \geqslant 0 \\ \langle \bar{\boldsymbol{\mu}}, \boldsymbol{h}(\bar{\boldsymbol{\varphi}}) \rangle = 0 \end{cases}. \tag{2.12}$$

The three latter conditions relate to the design constraints. They include design constraint feasibility, dual design constraint feasibility, and a complementarity condition, respectively. For each constraint h_i and multiplier $\bar{\mu}_i$, these conditions guarantee that either the constraint is active so that $h_i(\bar{\boldsymbol{\varphi}}) = 0$ and $\bar{\mu}_i > 0$, or that the constraint is inactive and thus $h_i(\bar{\boldsymbol{\varphi}}) < 0$ and $\bar{\mu}_i = 0$. In the latter case, the constraint contribution disappears from the Lagrangian (2.11). A variety of different approaches exists to solve design constrained optimization problems for an optimal point $(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}, \bar{\boldsymbol{q}}^*, \bar{\boldsymbol{\mu}})$ (see e.g. [103]). The choice and adjustment of such an approach for the particularities of the magnetic divertor design problem are a challenge that will be discussed in chapter 4. In this chapter, the focus will be on introducing the existing methods for optimization problems with exclusively state constraints.

2.1.2 Gradient calculation

The forward approach to gradient calculation

To solve the optimization problem (2.1), first, a nested gradient based optimization algorithm can be considered [103]. This class of methods finds the direction of each optimization step based on evaluations of the gradient of the reduced objective functional $\nabla\hat{\mathcal{J}}$. If combined with an appropriate globalization strategy (discussed further in section 2.3), gradient based optimization algorithms offer a guaranteed descent on the objective. Using the chain rule, the directional derivative of $\hat{\mathcal{J}}$ evaluates as

$$\delta\hat{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi}) = \left\langle \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}), \boldsymbol{\delta\varphi} \right\rangle + \left\langle \nabla_{\boldsymbol{q}}\mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}), \boldsymbol{\delta q}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi}) \right\rangle, \tag{2.13}$$

with $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{\varphi})$, and $\boldsymbol{\delta q}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi})$ the directional derivative of $\boldsymbol{q}(\boldsymbol{\varphi})$in the direction $\boldsymbol{\delta\varphi}$. For ease of notation, the dependence $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{\varphi})$ and the arguments of the directional derivatives $\delta\hat{\mathcal{J}}$ and $\boldsymbol{\delta q}$ will be left out below whenever they are clear from the context. If the state equation $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) = 0$ has been solved for $\boldsymbol{q}(\boldsymbol{\varphi})$, linearizing the constraints $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) = 0$ in direction $\boldsymbol{\delta\varphi}$ yields that the directional sensitivity $\boldsymbol{\delta q}$ is the solution to

$$\partial_{\boldsymbol{q}}\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q})\, \boldsymbol{\delta q} = -\partial_{\boldsymbol{\varphi}}\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q})\, \boldsymbol{\delta\varphi}. \tag{2.14}$$

After discretization of equations (2.13) and (2.14) to a finite-dimensional setting, the gradient vector $\nabla\hat{\mathcal{J}}$ can be evaluated numerically via a solution of equation (2.14) for the directional derivative $\boldsymbol{\delta q}$ in a specific direction $\boldsymbol{\delta\varphi}$ for each of the $n_{\boldsymbol{\varphi}}$ components of $\nabla\hat{\mathcal{J}}$. Clearly, this can become expensive if the control variable vector $\boldsymbol{\varphi}$ has a large number of components. Hence, care has to be taken that gradient evaluations do not unnecessarily dominate the computational cost of the optimization algorithm.

The adjoint approach to gradient calculation

Adjoint sensitivity calculations offer an attractive alternative to the direct solution of equations (2.13) and (2.14), and a gradient evaluation cost that is independent of the number of (discretized) control variables (see e.g. [21, 73]). Starting from a solved state $\boldsymbol{q}(\boldsymbol{\varphi})$, one possible derivation substitutes the linearized relation (2.14) between flow and control variables into the chain rule (2.13). This gives

$$\delta\hat{\mathcal{J}} = \left\langle \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q}), \boldsymbol{\delta\varphi}\right\rangle - \left\langle \nabla_{\boldsymbol{q}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q}), \partial_{\boldsymbol{q}}\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})\boldsymbol{\delta\varphi}\right\rangle. \tag{2.15}$$

Using the definition of the adjoint operator in equation (2.9) for the derivative $\partial_{\boldsymbol{q}}\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})^{-1}$, this can also be formulated as[†]

$$\delta\hat{\mathcal{J}} = \left\langle \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q}), \boldsymbol{\delta\varphi}\right\rangle - \left\langle \partial_{\boldsymbol{q}}\boldsymbol{c}^{-*}(\boldsymbol{\varphi},\boldsymbol{q})\,\nabla_{\boldsymbol{q}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q}), \partial_{\boldsymbol{\varphi}}\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})\boldsymbol{\delta\varphi}\right\rangle. \tag{2.16}$$

Solving the adjoint equation

$$\partial_{\boldsymbol{q}}\boldsymbol{c}^{*}(\boldsymbol{\varphi},\boldsymbol{q})\,\boldsymbol{q}^{*} = -\nabla_{\boldsymbol{q}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q}) \tag{2.17}$$

for

$$\boldsymbol{q}^{*}(\boldsymbol{\varphi}) = -\partial_{\boldsymbol{q}}\boldsymbol{c}^{-*}(\boldsymbol{\varphi},\boldsymbol{q})\,\nabla_{\boldsymbol{q}}\mathcal{J}(\boldsymbol{\varphi},\boldsymbol{q})$$

and using the adjoint of the linearization with respect to the design variables, $\partial_{\boldsymbol{\varphi}}\boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q})^{*}$, now allows reformulating this as an explicit expression for the reduced objective gradient

$$\nabla\hat{\mathcal{J}}(\boldsymbol{\varphi}) = \nabla_{\boldsymbol{\varphi}}\mathcal{J}\,(\boldsymbol{\varphi},\boldsymbol{q}) + \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^{*}(\boldsymbol{\varphi},\boldsymbol{q})\,\boldsymbol{q}^{*}, \tag{2.18}$$

[†]The notation $A^{-*} := (A^{-1})^{*}$ is used here.

where $\partial_{\boldsymbol{\varphi}} \boldsymbol{c}^*(\boldsymbol{\varphi}, \boldsymbol{q})$ can often be obtained through an analytical differentiation. For the scalar-valued objective function it can be seen that this approach might significantly reduce the computational cost with respect to the direct evaluation of equations (2.13) and (2.14).

Alternatively, the adjoint approach to sensitivity calculation can be derived using the Lagrangian (2.7). First remark that

$$\delta\hat{\mathcal{J}} = \hat{\mathcal{L}}' = \partial_{\boldsymbol{\varphi}}\mathcal{L}\,\boldsymbol{\delta\varphi} + \partial_{\boldsymbol{q}}\mathcal{L}\,\boldsymbol{\delta q} + \partial_{\boldsymbol{q}^*}\mathcal{L}\,\boldsymbol{q}^{*\prime},$$

where the first equality trivially follows from the definition of the reduced Lagrangian $\hat{\mathcal{L}}(\boldsymbol{\varphi}) := \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}(\boldsymbol{\varphi}), \boldsymbol{q}^*(\boldsymbol{\varphi}))$ and $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}(\boldsymbol{\varphi})) \triangleq 0$. Then, observe that the last two terms of this equality are equal to zero at *primal* (state) and *dual* (adjoint) feasibility (see equation (2.8)). After subsequently solving these equations for $\boldsymbol{q}(\boldsymbol{\varphi})$ and $\boldsymbol{q}^*(\boldsymbol{\varphi})$, respectively, only the partial derivative of the Lagrangian with respect to the control variables $\boldsymbol{\varphi}$ is thus nonzero. The gradient $\nabla\hat{\mathcal{J}}$ then reads

$$\nabla\hat{\mathcal{J}}(\boldsymbol{\varphi}) = \nabla_{\boldsymbol{\varphi}}\mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*) \tag{2.19}$$

$$= \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}) + \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^*(\boldsymbol{\varphi}, \boldsymbol{q})\,\boldsymbol{q}^*, \tag{2.20}$$

which equals the expression found from the former derivation. Remark that setting (2.20) to zero then again leads to the design equation of section 2.1.1 in absence of design constraints.

Discretization and adjoint approaches

To solve the set of PDEs $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) = 0$ numerically, these equations need to be discretized to a discrete problem $\boldsymbol{c}_h(\boldsymbol{\varphi}_h, \boldsymbol{q}_h)$, with a discrete control vector $\boldsymbol{\varphi}_h$ and a spatially and/or temporally discretized state vector $\boldsymbol{q}_h$ and where the subscript h refers to a characteristic spacing distance of the grid. In the adjoint approach to sensitivity calculation as described above, a second set of PDEs is obtained, namely the adjoint PDEs (equation (2.17)). Likewise, these adjoint equations should be discretized before they can be evaluated numerically. Introducing the discretization operator $\{\cdot\}_{\mathcal{H}}$ that densely approaches elements of an

infinite-dimensional Hilbert space in a finite-dimensional subspace, the discretized adjoint equations

$$\left\{\nabla_{\boldsymbol{q}}\mathcal{J}\right\}_{\mathcal{H}} + \left\{\partial_{\boldsymbol{q}}\boldsymbol{c}^*\right\}_{\mathcal{H}}\boldsymbol{q}^*_{\mathcal{H}} = 0 \tag{2.21}$$

are obtained, with $\boldsymbol{q}^*_{\mathcal{H}} = \{\boldsymbol{q}^*\}_{\mathcal{H}}$. Similarly, equation (2.18) can be discretized to obtain a discrete approximation of the design sensitivity. This approach to sensitivity calculation is called the Optimize-Then-Discretize (OTD) or continuous adjoint approach.

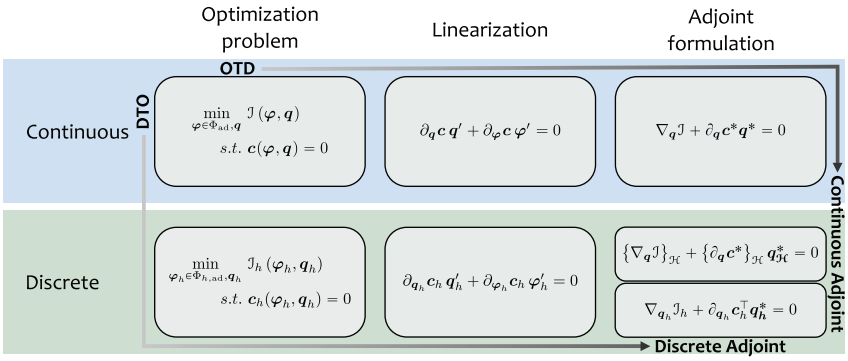


Figure 2.2: The Optimize-Then-Discretize (OTD) or in short continuous adjoint approach versus the Discretize-Then-Optimize (DTO) or discrete adjoint approach. Figure based on Refs. [41, 44, 59].

An alternative approach, the so-called Discretize-Then-Optimize (DTO) or discrete adjoint approach, starts from the discretized optimization problem

$$\begin{aligned} \min_{\boldsymbol{\varphi}_h\in\Phi_{h,\mathrm{ad}},\boldsymbol{q}_h} \quad & \mathcal{J}_h(\boldsymbol{\varphi}_h,\boldsymbol{q}_h) \\ s.t. \quad & \boldsymbol{c}_h(\boldsymbol{\varphi}_h,\boldsymbol{q}_h) = 0. \end{aligned}$$

Analogous to the continuous case, an adjoint approach to sensitivity calculation can now be elaborated. In this derivation, the use of standard discrete ℓ_2 inner products

$$\langle \boldsymbol{x},\boldsymbol{y}\rangle = \boldsymbol{x}^\top\boldsymbol{y}. \tag{2.22}$$

will immediately be assumed. Again, the discrete state equations are solved first for $\boldsymbol{q}_h(\boldsymbol{\varphi}_h)$. The linearized discrete state equations

$$\partial_{\boldsymbol{q}_h} \boldsymbol{c}_h \delta \boldsymbol{q}_h + \partial_{\boldsymbol{\varphi}_h} \boldsymbol{c}_h \delta \boldsymbol{\varphi}_h = 0 \tag{2.23}$$

can then be used in complement with the chain rule for the sensitivity

$$\delta \hat{\mathcal{J}}_h = \partial_{\boldsymbol{\varphi}_h} \mathcal{J}_h \delta \boldsymbol{\varphi}_h + \partial_{\boldsymbol{q}_h} \mathcal{J}_h \delta \boldsymbol{q}_h, \tag{2.24}$$

to derive the set of discrete adjoint equations

$$\nabla_{\boldsymbol{q}_h} \mathcal{J}_h + \partial_{\boldsymbol{q}_h} \boldsymbol{c}_h^\top \boldsymbol{q}_{\boldsymbol{h}}^* = 0 \tag{2.25}$$

that are solved for $\boldsymbol{q}_{\boldsymbol{h}}^*(\boldsymbol{\varphi}_h)$ and reduce the design sensitivity calculation to

$$\nabla \hat{\mathcal{J}}_h = \nabla_{\boldsymbol{\varphi}_h} \mathcal{J}_h + \partial_{\boldsymbol{\varphi}_h} \boldsymbol{c}_h^\top \boldsymbol{q}_{\boldsymbol{h}}^*. \tag{2.26}$$

In this equation, it can be easily seen that the cost of the gradient evaluation now reduces to that of a matrix-vector product.

The difference between the discrete and continuous adjoint approach to sensitivity calculation is shown schematically in figure 2.2. It is important to note that they are not necessarily equivalent, since the continuous adjoint approach allows for different discretization choices for state and adjoint equations. If, by careful choice of inner product and discretization, the DTO and OTD approach are equivalent, dual consistency has been achieved. In principle, the difference should be mainly originating from discretization choices and the approaches should therefore both converge to the solution in infinite-dimensional space for infinitesimally small grid cell sizes.

Advantages of the discrete and continuous adjoint methods have been discussed extensively in literature [41, 57, 59, 100, 101, 137]. The most notable advantage of the discrete adjoint approach clearly being the consistency between discretized model and gradient that allows full convergence to the discrete optimum. Using the continuous adjoint, gradients close to a minimum might no longer point in a descent direction due to this inconsistency, stalling convergence. Nevertheless, deriving the discrete adjoint equations by hand is more cumbersome

than its continuous adjoint equivalent. Therefore, so-called Algorithmic Differentiation (AD) tools are often exploited. Unfortunately, these methods often come at an increased memory consumption and often also increased computational cost. Nevertheless, new code developments are easily incorporated without additional adjoint derivations. Similarly, it does not require deriving new adjoint boundary conditions whenever the forward boundary conditions are changed. The flexibility of this AD approach often outweighs the other arguments.

Another advantage of the discrete adjoint method is the ease of the sensitivity verification with finite difference gradients. In contrast, the continuous adjoint gradient can only be verified through systematic grid refinement. However, straightforward application of the discrete adjoint method without consideration of the dependencies in the continuous model and the continuity of their discrete implementation might lead to inaccurate sensitivities. Although many more arguments can be mentioned here, this short discussion already indicates that both methods have their own advantages and drawbacks. In fact, one might even resort to using hybrid continuous-discrete adjoint methods such as introduced in Ref. [137], to exploit the advantages of both methods for different model equations in a coupled system of equations.

2.2 Optimization algorithms

The PDE-constrained optimization problems under consideration are mostly highly nonlinear optimization problems. Such problems thus need, in most practical applications, an iterative solver to obtain an optimum of the KKT-conditions introduced in section 2.1.1. Two major categories of adjoint based optimization methods can be used.

The first category includes the nested optimization methods. These methods solve state and adjoint equations in each optimization step to find the cost function value $\hat{\mathcal{J}}$ and the gradient $\nabla\hat{\mathcal{J}}$ and use them to calculate a design step $\Delta\boldsymbol{\varphi}$. This approach successfully reuses the forward PDE-simulation as is, which is of primary importance since these codes are often very large and result from specific solver development for the particular application. Nested optimization methods then build an external optimization layer around the solver (and adjoint solver).

The relative simplicity being the main advantage of the approach. The obvious disadvantage being that much CPU-time is 'waisted' converging the state and adjoint equations until feasibility in each optimization iteration.

The second category includes methods that simultaneously solve for all KKT-conditions, both feasibility and optimality. These two different types of optimization approaches are briefly introduced here. For a more complete overview of methods for PDE-constrained optimization the reader is referred to [21]. For ease of notation, all terms in the remainder of this section will be either considered discrete or discretized, without explicitly writing the subscript $_h$ or the discretization operator $\{\cdot\}_{\mathcal{H}}$.

2.2.1 Nested optimization methods

In absence of design constraints, finding a solution to the KKT system through nested optimization methods reduces to iteratively looking for optimal design variables $\bar{\boldsymbol{\varphi}}$ at which $\nabla\hat{\mathcal{J}} = 0$. A variety of methods exist, amongst which the steepest descent method is the simplest. This method updates the design $\boldsymbol{\varphi}_k$ in the k^{th} optimization iteration by making a step in the negative gradient direction as $\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - \alpha\nabla\hat{\mathcal{J}}$, using a constant scalar multiplier α. At the cost of increased storage, convergence can often be accelerated by using a quasi-Newton method. This method additionally approximates the *reduced Hessian* $B \approx \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}}\hat{\mathcal{J}}$ and then updates the design variables using the approximate Newton step $\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - B^{-1}\nabla\hat{\mathcal{J}}$. B is called the reduced Hessian since it is based on the Hessian of the reduced cost functional, from which the state variables $\boldsymbol{q}$ are eliminated using the state equations. B is also known here as the design preconditioner, since it serves as a preconditioner for the steepest descent step.

Since Hessian evaluations are expensive, B is estimated using exclusively gradient data from previous optimization iterations. The well-known Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm performs in each optimization iteration the rank-two Hessian update

$$B_{k+1} = B_k - \frac{B_k \boldsymbol{s_k} \boldsymbol{s_k}^\top B_k}{\boldsymbol{s_k}^\top B_k \boldsymbol{s_k}} + \frac{\boldsymbol{y_k} \boldsymbol{y_k}^\top}{\boldsymbol{y_k}^\top \boldsymbol{s_k}}, \tag{2.27}$$

with $\boldsymbol{s_k} = \boldsymbol{\varphi}_{k+1} - \boldsymbol{\varphi}_k$ and $\boldsymbol{y_k} = \nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_{k+1}) - \nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_k)$. If storing the entire Hessian is too memory-intensive, the Limited-memory BFGS (L-BFGS) offers an alternative [103]. When working with an unconstrained optimization problem, it might moreover be convenient to use the Sherman-Morrison-Woodbury formula to find a similar expression for the inverse design preconditioner $H = B^{-1}$ (see e.g. [103]).

It is essential for the algorithm that B is sufficiently positive definite to have bounded design steps in the direction of a minimum. A first option is to use damped BFGS updates. This update rule relaxes the Hessian update to guarantee the directional curvature $\boldsymbol{s_k}^\top B_k \boldsymbol{s_k}$ can maximally be decreased by a fraction γ in the novel update [112]. More specifically, it applies the following rule: If $\boldsymbol{s_k}^\top \boldsymbol{y_k} < \gamma\, \boldsymbol{s_k}^\top B_k \boldsymbol{s_k}$, substitute $\boldsymbol{y_k}$ by $\tilde{\boldsymbol{y}}_{\boldsymbol{k}} = \theta \boldsymbol{y_k} + (1-\theta) B_k \boldsymbol{s_k}$, such that $\boldsymbol{s_k}^\top \tilde{\boldsymbol{y}}_{\boldsymbol{k}} = \gamma\, \boldsymbol{s_k}^\top B_k \boldsymbol{s_k}$. This is achieved when

$$\theta = \frac{(1-\gamma)\boldsymbol{s_k}^\top B_k \boldsymbol{s_k}}{\boldsymbol{s_k}^\top B_k \boldsymbol{s_k} - \boldsymbol{s_k}^\top \boldsymbol{y_k}}. \tag{2.28}$$

The resulting $\tilde{\boldsymbol{y}}_{\boldsymbol{k}}$ then simply replaces $\boldsymbol{y_k}$ in equation (2.27). Alternatively, one can make sure in a line search algorithm that the accepted step obeys the Wolfe conditions, as presented in section 2.3.

2.2.2 SQP, rSQP and one-shot methods

As an alternative to the nested optimization methods, one may aim at simultaneously attaining feasibility and optimality. A direct Newton solve on the (state-constrained) KKT system (2.8) directly leads to the Sequential Quadratic Programming (SQP) approach that solves for a change in costate $\Delta\boldsymbol{q}^*$, design $\Delta\boldsymbol{\varphi}$, and state $\Delta\boldsymbol{q}$ as

$$\begin{bmatrix} \nabla_{\boldsymbol{qq}}\mathcal{L} & \nabla_{\boldsymbol{q\varphi}}\mathcal{L} & \partial_{\boldsymbol{q}}\boldsymbol{c}^* \\ \nabla_{\boldsymbol{\varphi q}}\mathcal{L} & \nabla_{\boldsymbol{\varphi\varphi}}\mathcal{L} & \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^* \\ \partial_{\boldsymbol{q}}\boldsymbol{c} & \partial_{\boldsymbol{\varphi}}\boldsymbol{c} & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{q} \\ \Delta\boldsymbol{\varphi} \\ \Delta\boldsymbol{q}^* \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{q}}\mathcal{L} \\ -\nabla_{\boldsymbol{\varphi}}\mathcal{L} \\ -\boldsymbol{c} \end{bmatrix}, \tag{2.29}$$

where one may note that in a discrete adjoint approach or a dually consistent continuous adjoint approach the matrix on the left hand side is symmetric, since $\partial_{\boldsymbol{q}}\boldsymbol{c}^*$ and $\partial_{\boldsymbol{\varphi}}\boldsymbol{c}^*$ may be replaced by $\partial_{\boldsymbol{q}}\boldsymbol{c}^\top$ and $\partial_{\boldsymbol{\varphi}}\boldsymbol{c}^\top$, respectively.

Often, it is more interesting to use the reduced SQP (rSQP) formulation

$$\begin{bmatrix} 0 & 0 & \partial_{\boldsymbol{q}}\boldsymbol{c}^* \\ 0 & B & \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^* \\ \partial_{\boldsymbol{q}}\boldsymbol{c} & \partial_{\boldsymbol{\varphi}}\boldsymbol{c} & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{q} \\ \Delta\boldsymbol{\varphi} \\ \Delta\boldsymbol{q}^* \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{q}}\mathcal{L} \\ -\nabla_{\boldsymbol{\varphi}}\mathcal{L} \\ -\boldsymbol{c} \end{bmatrix}, \tag{2.30}$$

since the second derivatives in (2.29) are often too expensive to calculate. The Hessian B can be seen as a result of projecting the Hessian of the Lagrangian onto the nullspace of the model constraints [21, p. 55 and p. 62]. That is, consider the Hessian of the quadratic program

$$H := \begin{bmatrix} \nabla_{\boldsymbol{q}\boldsymbol{q}}\mathcal{L} & \nabla_{\boldsymbol{q}\boldsymbol{\varphi}}\mathcal{L} \\ \nabla_{\boldsymbol{\varphi}\boldsymbol{q}}\mathcal{L} & \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}}\mathcal{L} \end{bmatrix} \tag{2.31}$$

and define

$$Z := \begin{bmatrix} \partial_{\boldsymbol{\varphi}}\boldsymbol{q} \\ \mathbb{I} \end{bmatrix} = \begin{bmatrix} -\partial_{\boldsymbol{q}}\boldsymbol{c}^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c} \\ \mathbb{I} \end{bmatrix}. \tag{2.32}$$

The linearized model equations can then again be used to eliminate the dependence $\boldsymbol{q}(\boldsymbol{\varphi})$ and obtain the reduced Hessian

$$\begin{aligned} B = Z^\top H Z = \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}}\mathcal{L} - \nabla_{\boldsymbol{\varphi}\boldsymbol{q}}\mathcal{L}\ \partial_{\boldsymbol{q}}\boldsymbol{c}^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c} - \left(\partial_{\boldsymbol{q}}\boldsymbol{c}^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c}\right)^\top \nabla_{\boldsymbol{q}\boldsymbol{\varphi}}\mathcal{L} + \\ \left(\partial_{\boldsymbol{q}}\boldsymbol{c}^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c}\right)^\top \nabla_{\boldsymbol{q}\boldsymbol{q}}\mathcal{L}\ \partial_{\boldsymbol{q}}\boldsymbol{c}^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c}. \end{aligned} \tag{2.33}$$

Here, the reduced Hessian size can be efficiently exploited using a quasi-Newton update technique as introduced in section 2.2.1. Additionally, the lower triangular structure of the rSQP-matrix is very suitable for sequential solution by elimination.

Yet, the complete constraint Jacobian is often not available in a typical PDE-simulation code. Because of the large state space, iterative solvers are typically used that approximate the constraint derivative $\partial_{\boldsymbol{q}}\boldsymbol{c}$ with a more convenient approximation A, possibly including additional relaxation or a false time-stepping term. Using this approximation in the rSQP method leads to an approximate rSQP method

$$\begin{bmatrix} 0 & 0 & A^* \\ 0 & B & \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^* \\ A & \partial_{\boldsymbol{\varphi}}\boldsymbol{c} & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{q} \\ \Delta\boldsymbol{\varphi} \\ \Delta\boldsymbol{q}^* \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{q}}\mathcal{L} \\ -\nabla_{\boldsymbol{\varphi}}\mathcal{L} \\ -\boldsymbol{c} \end{bmatrix}. \tag{2.34}$$

This approximate rSQP system can now be solved through elimination with a Gauss-Seidel iteration or by further approximating this system as

$$\begin{bmatrix} 0 & 0 & A^* \\ 0 & B & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{q} \\ \Delta \boldsymbol{\varphi} \\ \Delta \boldsymbol{q}^* \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{q}} \mathcal{L} \\ -\nabla_{\boldsymbol{\varphi}} \mathcal{L} \\ -\boldsymbol{c} \end{bmatrix}. \tag{2.35}$$

and using a parallel Jacobi-type iteration. These latter two types of iterative approximate rSQP solvers that integrally reuse the existing state solver are so-called single step one-shot methods, since they perform one iteration on the state solver, one on the adjoint solver and perform one design update. The parallel Jacobi-type iteration is used mostly if algorithmic differentiation is used, since function value and design gradient are calculated simultaneously. Since these one-shot methods—like nested methods—allow leaving the structure of the forward solver unchanged, they are of special interest for typical PDE-constrained applications. The difference between nested and one-shot optimization methods is schematically shown in figure 2.3.

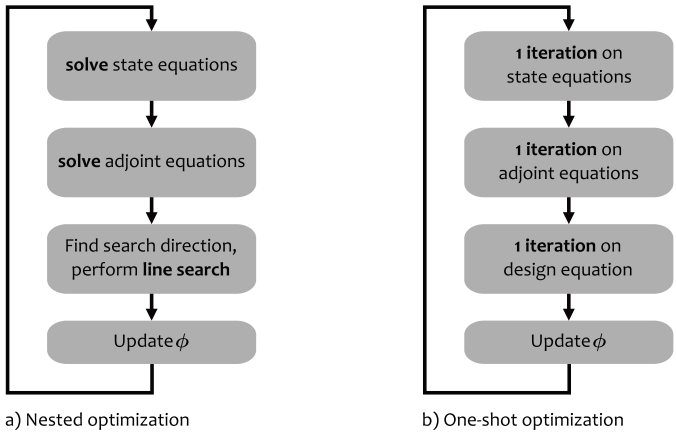


Figure 2.3: A schematic illustration of the difference between a) nested optimization methods and b) one-shot methods.

2.3 Globalization

In this section, the concept of globalization will be briefly introduced for nested optimization. For one-shot methods, globalization is a relatively new topic, further discussed in chapter 7. Newton-type methods only converge if the initial solution is sufficiently close to the optimum. To achieve global convergence, a globalization strategy can then be introduced. It should be stressed that global convergence means converging from any initial starting point to a (possibly local) stationary point. Typically, this is achieved by changing the design step such that the cost functional is decreased in each optimization step.

Two important categories of globalization methods exist. Trust region methods and line search methods. Although both methods are effective and have their own advantages, only line search methods will be discussed in this PhD thesis. The choice for these methods mainly stems from the decoupling between search direction calculation and globalization, leading to increased code transparency.

After the search direction calculation, an inexact line search procedure searches along the resulting direction $\boldsymbol{\delta\varphi}_k$ for a point $\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k + \tau_k \boldsymbol{\delta\varphi}_k$ with sufficient descent on the objective function. Typically, until a point is found that satisfies the Armijo condition

$$\hat{\mathcal{J}}(\boldsymbol{\varphi}_k + \tau_k \boldsymbol{\delta\varphi}_k) \leqslant \hat{\mathcal{J}}(\boldsymbol{\varphi}_k) + c_1 \tau_k \left(\nabla \hat{\mathcal{J}}(\boldsymbol{\varphi}_k), \boldsymbol{\delta\varphi}_k \right), \tag{2.36}$$

with $c_1 \in (0, \frac{1}{2})$ a parameter relaxing the gradient. Additionally, the curvature condition

$$\left| \left(\nabla \hat{\mathcal{J}}(\boldsymbol{\varphi}_k + \tau_k \boldsymbol{\delta\varphi}_k), \boldsymbol{\delta\varphi}_k \right) \right| \leqslant c_2 \left| \left(\nabla \hat{\mathcal{J}}(\boldsymbol{\varphi}_k), \boldsymbol{\delta\varphi}_k \right) \right| \tag{2.37}$$

can be imposed, with $c_2 \in (0, 1)$ a parameter indicating how much the curvature is to be reduced in each step. These two conditions together form the so-called Wolfe conditions. Although gradient calculation is required at each line search point when using the second Wolfe condition, it has several advantages. Firstly, it inherently avoids that negative curvatures are found when using a quasi-Newton approach. Secondly, it sets out a minimal step size so that in general less optimization iterations are needed. For more information on globalization approaches the reader is referred to Ref. [103].

2.4 Summary

In this chapter, the optimization problem, its characteristics, and the KKT conditions that govern its solutions were described first. Further, the direct and adjoint approaches to sensitivity calculation were described. While the computational cost of a direct sensitivity evaluation scales strongly with the size of the design vector, the adjoint approach calculates the sensitivity of a scalar-valued cost functional efficiently. Moreover, it is pointed out that the choice between using the adjoint on the continuous or discretized optimization problem has important consequences.

Subsequently, a brief overview of solution methods was presented to solve the optimization problem. Nested quasi-Newton type optimization methods and approximate rSQP type one-shot methods were indicated as potentially interesting options, since the original simulation code can be augmented with the optimization functionalities in a non-intrusive way. Yet, incorporation of design constraints still needs consideration. Finally, globalization was discussed for nested optimization algorithms.

In this thesis, both nested and one-shot algorithms will be examined. The application of the former in the context of magnetic divertor design is treated in chapter 4. The latter is elaborated in chapter 7. There, the introduction of a one-shot globalization approach will also be discussed in detail.

3

Integral Magnetic Divertor Model

In nuclear fusion tokamaks, plasma, target plates, and magnetic configuration interact in a complex way. Yet, in current models, magnetic field and plasma edge flow are treated independently. That is, plasma edge transport codes calculate particle, momentum and energy flows on a precomputed and then fixed magnetic field. This approximation is justified by the relatively small plasma current in divertor and SOL.

Since the objective of this thesis is the automated design of the magnetic configuration to improve the heat exhaust, the subject of this first chapter therefore comprises the development of an integrated model that governs both the magnetics and the plasma edge transport of a tokamak in steady state. Furthermore, both models should be incorporated into an automated code that is able to evaluate the heat load, starting from the divertor coils that govern the magnetic divertor configuration. A flow chart of such a procedure is presented in figure 3.1. It should be noted that also the grid generation process is an inherent component of this integrated code. This is due to the strongly anisotropic flow, which is orders of magnitude greater along than perpendicular to the magnetic field. Consequently, the grid on which the plasma edge transport equations are discretized should be aligned to the magnetic field to avoid excessive numerical diffusion. One of the challenges is therefore the development and inclusion of an automated grid generator.

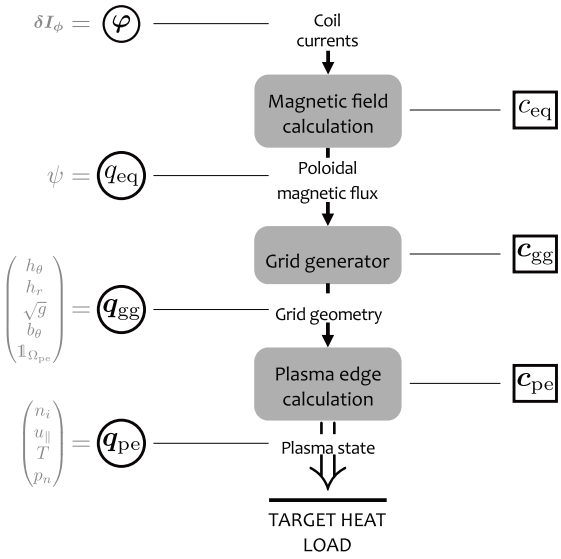


Figure 3.1: The components of the forward simulation

A range of models with different levels of sophistication could be selected to simulate the magnetic field and plasma edge flows. However, since the main goal is the development of the optimal design techniques, we choose slightly reduced models that are easy enough to test the concepts, but retain the principal dependencies. At a later stage in the design process, one might replace both the magnetic field and the plasma edge transport model with state-of-the-art simulation software and use these very same or slightly adjusted design methods on the more extended models. By doing so, an improved design might be achieved, still bearing in mind that model deficiencies might influence the optimal solution.

In this chapter, we will treat the model components depicted in figure 3.1 one by one. For uniformity, we will consider both the continuous model and its discretization in this chapter. In the first section of this chapter a very fast and simple model is presented to account for small changes to a magnetic equilibrium. Next, in section 3.2, the coordinate transformation that governs the grid generation process is outlined as well as its discrete equivalent. Since plasma edge grid generators often heavily rely on user interaction, the automation of this grid

generation process is briefly discussed here. Finally, in section section 3.3, the plasma edge transport model that evaluates the target heat load is presented.†

3.1 Magnetic field modelling

3.1.1 Quasi-static magnetic field equations

Magnetic fields are governed by Maxwell's equations. This is a set of 4 PDEs of which two are of direct relevance for steady-state magnetic fields. The first one, also known as Gauss' law for magnetism, states that magnetic fields should be divergence free or equivalently, that magnetic monopoles do not exist. Mathematically this is expressed by the differential equation

$$\nabla \cdot \boldsymbol{B} = 0, \tag{3.1}$$

with $\boldsymbol{B}$ the magnetic field density vector in [T]. The second condition is Ampère's law for quasi-static fields,

$$\nabla \times \frac{\boldsymbol{B}}{\mu} = \boldsymbol{J}, \tag{3.2}$$

with $\boldsymbol{J}$ the current density vector in $\left[\frac{\mathrm{A}}{\mathrm{m}^2}\right]$.

These laws can then be combined by writing Ampères law in terms of a vector potential $\boldsymbol{A}$, defined by

$$\boldsymbol{B} := \nabla \times \boldsymbol{A} \tag{3.3}$$

Gauss' law for magnetic fields (3.1) is then trivially fulfilled and Ampères law then reads

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \boldsymbol{A}\right) = \boldsymbol{J}. \tag{3.4}$$

However, the vector potential $\boldsymbol{A}$ is not uniquely defined, since adding curl-free components to the potential will not change the magnetic field. The Coulomb

†Parts of this chapter have been published in "Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N. & Reiter, D. (2014). An Automated Approach to Magnetic Divertor Configuration Design. *Nuclear Fusion*, **55**" and "Blommaert, M., Heumann, H., Baelmans, M., Gauger, N.R. & Reiter, D. (2016). Towards Automated Magnetic Divertor Design for Optimal Heat Exhaust. *ESAIM: Proceedings and Surveys*, **53**, 49–63".

gauge ($\nabla \cdot \boldsymbol{A} = 0$) is therefore used to uniquely define $\boldsymbol{A}$. Using this gauge and the vector identity

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \boldsymbol{A}\right) = \nabla\left(\frac{1}{\mu}\nabla \cdot \boldsymbol{A}\right) - \nabla \cdot \left(\frac{1}{\mu}\nabla \boldsymbol{A}\right),$$

equation (3.4) can be simplified to

$$-\nabla \cdot \left(\frac{1}{\mu}\nabla \boldsymbol{A}\right) = \boldsymbol{J}. \tag{3.5}$$

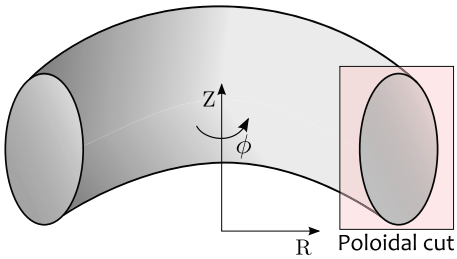


Figure 3.2: Illustration of the $(\mathrm{R}, \phi, \mathrm{Z})$-coordinate system in a tokamak, and of a poloidal plane in this coordinate system.

3.1.2 Exploiting toroidal (quasi-)symmetry

In steady-state tokamak analysis, toroidal symmetry is often assumed. Consider a cylindrical $(\mathrm{R}, \phi, \mathrm{Z})$-coordinate system as depicted in figure 3.2, with unit vectors $\mathbf{e}_\mathrm{R}, \mathbf{e}_\phi$ and $\mathbf{e}_\mathrm{Z}$. The poloidal magnetic field $(B_\mathrm{R}\,\mathbf{e}_\mathrm{R} + B_\mathrm{Z}\,\mathbf{e}_\mathrm{Z})$ in axisymmetric configurations is entirely determined by A_ϕ through the toroidal component of equation (3.5),

$$\left[-\nabla \cdot \left(\frac{1}{\mu}\nabla \boldsymbol{A}\right)\right]_\phi = J_\phi. \tag{3.6}$$

In terms of the poloidal magnetic flux per radian ψ, defined as $\psi = \mathrm{R}A_\phi$, the governing equation

$$-\mathrm{R}\,\nabla \cdot \left(\frac{1}{\mu \mathrm{R}^2}\nabla \psi\right) = J_\phi \tag{3.7}$$

is finally obtained. In vacuum, this equation reduces further to

$$\Delta^* \psi = -\mu_0 J_\phi, \tag{3.8}$$

with μ_0 the magnetic permeability of vacuum (in absence of dia-, para- or ferromagnetic structures) and the elliptic operator Δ^* defined as:

$$\Delta^* \psi = \mathrm{R} \frac{\partial}{\partial \mathrm{R}} \left(\frac{1}{\mathrm{R}} \frac{\partial \psi}{\partial \mathrm{R}} \right) + \frac{\partial^2 \psi}{\partial \mathrm{Z}^2}. \tag{3.9}$$

The magnetic field components B_R an B_Z can be retrieved from the poloidal flux ψ as[†]

$$B_\mathrm{R} = -\frac{1}{\mathrm{R}} \frac{\partial \psi}{\partial \mathrm{Z}} \qquad \text{and} \qquad B_\mathrm{Z} = \frac{1}{\mathrm{R}} \frac{\partial \psi}{\partial \mathrm{R}}. \tag{3.10}$$

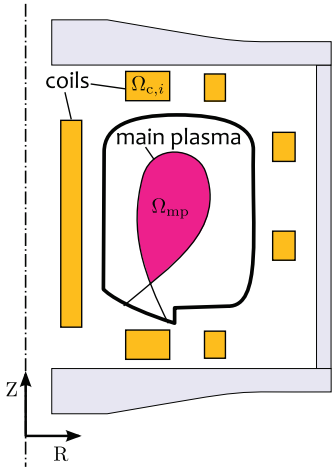


Figure 3.3: A conceptual figure of a poloidal tokamak cut. The domain covered by coils $\Omega_{c,i}$ and the main plasma domain Ω_{mp} are indicated in yellow and pink, respectively.

[†]Remark that in literature instead of the poloidal flux per radian ψ sometimes the total flux integrated over the toroidal direction $\Psi = 2\pi\psi$ is used, leading to regular mismatches by a factor 2π, as experienced by the author.

3.1.3 Towards a free boundary equilibrium problem

In tokamaks, the magnetic field is constituted by the coils around the vacuum chamber as well as the field induced by the plasma currents themselves. The expression for the toroidal current density J_ϕ is therefore region dependent (see figure 3.3). Within coil number i with domain $\Omega_{\mathrm{c},i}$ and cross-section S_i, the current density is simply given by $I_{\phi,i}/S_i$, with $I_{\phi,i}$ the coil current. In the main plasma domain Ω_{mp} bounded by the last closed flux surface (LCFS), the magnetic flux ψ is approximately described by the *Grad-Shafranov* (GS) equation for *ideal static magnetohydrodynamic* (MHD) *equilibrium* [60, 129]

$$\Delta^*\psi\left(\mathrm{R},\mathrm{Z}\right) = -\mu_0\mathrm{R}^2 p^{'} - FF^{'}, \tag{3.11}$$

with p and F flux functions associated to the pressure and the R-scaled toroidal magnetic field component, respectively, and where the accent denotes a differentiation with respect to the poloidal flux ψ. The 'static' assumption in this equilibrium equation reduces the complexity of the original problem by assuming that the plasma particle velocities are negligible. The remaining equation (3.11) balances Lorentz force against plasma pressure. Outside the main plasma domain Ω_{mp}, plasma currents are typically neglected.

Rewriting the GS equation in terms of the toroidal plasma current density J_P, the governing equations thus read [18]

$$\begin{aligned} &L\psi = J_\phi \qquad\qquad \text{, with} \\ &J_\phi = \begin{cases} J_\mathrm{P} = \mathrm{R}p^{'}(\psi) + 1/(\mu_0\mathrm{R})\; FF^{'}(\psi) & \text{in } \Omega_{\mathrm{mp}}\left(\psi\right), \\ I_{\phi,i}/S_i & \text{in } \Omega_{\mathrm{c},i}, i = 1\ldots n_c, \\ 0 & \text{elsewhere,} \end{cases} \end{aligned} \tag{3.12}$$

where $I_{\phi,i}$ represents the current in the i^{th} toroidal coil, n_c represents the total number of coils, and where we introduce the operator

$$\begin{aligned} L(\cdot) &= -\mathrm{R}\,\nabla\cdot\left(\frac{1}{\mu(\psi)\mathrm{R}^2}\nabla(\cdot)\right) \\ &= -\frac{\partial}{\partial\mathrm{R}}\left(\frac{1}{\mu(\psi)\mathrm{R}}\frac{\partial(\cdot)}{\partial\mathrm{R}}\right) - \frac{\partial}{\partial\mathrm{Z}}\left(\frac{1}{\mu(\psi)\mathrm{R}}\frac{\partial(\cdot)}{\partial\mathrm{Z}}\right) - \frac{\partial}{\partial\phi}\left(\frac{1}{\mu(\psi)\mathrm{R}^3}\frac{\partial(\cdot)}{\partial\phi}\right) \end{aligned} \tag{3.13}$$

for ease of notation. Notice that the last term in equation (3.13) is zero because of toroidal symmetry. The solution of this problem is non-trivial, since this PDE is nonlinear in ψ through both the plasma current density J_P and the main plasma domain Ω_mp. Because of the latter, these equations are also referred to as the free boundary equilibrium (FBE) equations. Additionally, the permeability μ is a nonlinear function of ψ in ferro-magnetic structures.

3.1.4 A simple and fast perturbation approach

Because of the problem complexity, a perturbation model is used in a first step. In this perturbation model, it is assumed that magnetic field changes due to small coil perturbations can be sufficiently described using the *vacuum approximation*. This approximates the magnetic field resulting from external coil changes as if they where coils in vacuum. The influence of altering plasma currents and their position is hereby neglected for the small changes concerned. The vacuum approximation is used frequently in edge transport models for the study of resonant magnetic perturbations [52, 94, 125]. The validity of the vacuum approximation for this application is subject of discussion in [113] and will be further discussed in chapter 6 for its application to magnetic divertor design.

To explain the methodology further, the equations (3.12) are formally combined as

$$L\psi = \sum_{i=1}^{n_c} \frac{I_{\phi,i}}{S_i} \mathbb{1}_{\Omega_{\mathrm{c},i}} + J_\mathrm{P}(\psi)\mathbb{1}_{\Omega_\mathrm{mp}(\psi)}, \tag{3.14}$$

using the indicator functions

$$\mathbb{1}_{\Omega_{\mathrm{c},i}}(\mathbf{r}) := \begin{cases} 1 & \text{if } \mathbf{r} \in \Omega_{\mathrm{c},i} \\ 0 & \text{if } \mathbf{r} \notin \Omega_{\mathrm{c},i} \end{cases} \quad \text{and} \tag{3.15}$$

$$\mathbb{1}_{\Omega_\mathrm{mp}(\psi)}(\mathbf{r}) := \begin{cases} 1 & \text{if } \mathbf{r} \in \Omega_\mathrm{mp}(\psi) \\ 0 & \text{if } \mathbf{r} \notin \Omega_\mathrm{mp}(\psi) \end{cases} \tag{3.16}$$

to indicate whether a position vector $\mathbf{r}$ is included in the respective domain. Now consider an unperturbed magnetic equilibrium ψ_0 of a divertor configuration with coil currents $I_{\phi,i,0}$ and a corresponding plasma current density distribution $J_{\mathrm{P},0}$, which satisfy the magnetic equilibrium (3.12). The effect of small coil current

changes $\delta I_{\phi,i}$ can be described as a perturbation $\delta\psi$ to ψ_0,

$$\begin{aligned} L\psi =& L(\psi_0 + \delta\psi) \approx L\psi_0 - \frac{1}{\mu_0 \mathrm{R}}\Delta^*\delta\psi \\ =& \sum_{i=1}^{n_c} \frac{I_{\phi,i,0}}{S_i}\mathbb{1}_{\Omega_{\mathrm{c},i}} + \sum_{i=1}^{n_c} \frac{\delta I_{\phi,i}}{S_i}\mathbb{1}_{\Omega_{\mathrm{c},i}} + \\ & J_\mathrm{P}(\psi_0)\mathbb{1}_{\Omega_{\mathrm{mp}}(\psi_0)} + \delta J_\mathrm{P}(\psi)\max(\mathbb{1}_{\Omega_{\mathrm{mp}}(\psi_0)}, \mathbb{1}_{\Omega_{\mathrm{mp}}(\psi)}). \end{aligned} \tag{3.17}$$

The operator L is nonlinear due to the presence of the permeability $\mu(\psi)$. Yet, the approximation in (3.17) is justified by the fact that $\mu \approx \mu_0$ in a large part of the reactor. If iron components are present, the approximation is still reasonable if the iron is close to saturation. However, the changes of the plasma currents $\delta J_\mathrm{P}(\psi)$ are still hard to calculate, as they are indirectly dependent on the external currents $I_{\phi,i}$ through induced shifts in the flux functions p and F. The perturbation model then neglects the plasma current change $\delta J_\mathrm{P}(\psi)$ as well as all permeabilities $\mu \neq \mu_0$. The main advantage being that only the linear equation

$$\Delta^*\delta\psi = -\mu_0 \mathrm{R} \sum_{i=1}^{n_c} \frac{\delta I_{\phi,i}}{S_i}\mathbb{1}_{\Omega_{\mathrm{c},i}} \tag{3.18}$$

needs to be solved for the poloidal flux perturbation $\delta\psi$. The approach is illustrated in figure 3.4.

It important to note that in this perturbation approach, force balance, and hence Equation (3.11), will in general no longer strictly apply if $\delta\psi \neq 0$. It can however be reasoned that the vacuum approximation becomes more accurate when the changes to the magnetic flux $\delta\psi$ are limited to the divertor and plasma edge region, since the plasma currents are approximately zero there. This condition is for example realized if two divertor shaping coils are close to each other and their current changes δI_ϕ have opposed directions and similar magnitude. The poloidal flux change $\delta\psi$ due to these coils then decays very quickly in space.

To speed up magnetic field calculations even further, the additivity of the linear operator Δ^* can be used to calculate the magnetic field contribution for each coil separately. Assuming an infinitesimally thin toroidal conductor at (r', z') and making use of the vacuum approximation, one can derive an explicit

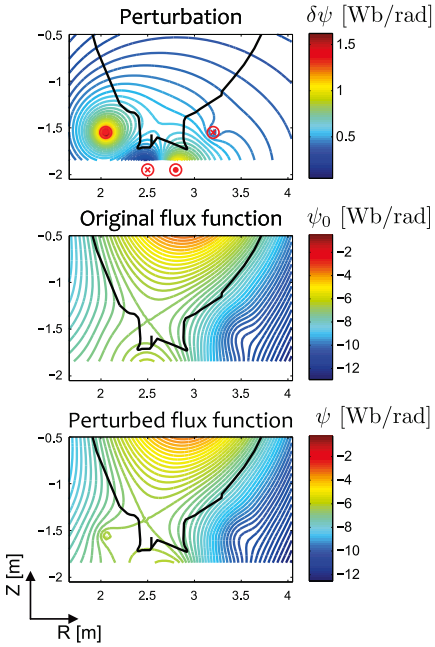


Figure 3.4: A perturbed magnetic flux ψ (bottom) is evaluated by adding a calculated perturbation $\delta\psi$ (top) onto an initial equilibrium ψ_0(middle). Notice the local changes in the divertor region of the perturbed magnetic flux compared to the reference magnetic flux.

expression for the magnetic flux perturbation $\delta\psi_i$ due to one divertor coil current change $\delta I_{\phi,i}$ [48]:

$$\delta\psi_i\left(\mathrm{R},\mathrm{Z},r',z'\right)=\frac{-\mu_0\mathrm{R}\,\delta I_{\phi,i}}{k\pi}\sqrt{\frac{r'}{\mathrm{R}}}\left[\left(1-\frac{k^2}{2}\right)K(k)-E(k)\right], \tag{3.19}$$

where

$$k^2=\frac{4\mathrm{R}r'}{\left(\mathrm{R}+r'\right)^2+\left(\mathrm{Z}-z'\right)^2} \tag{3.20}$$

and K and E are complete elliptical integrals of the first and second kind. These

are respectively given by

$$K(k) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}}, \tag{3.21}$$

$$E(k) = \int_0^{\frac{\pi}{2}} \sqrt{1 - k^2 \sin^2 \theta}\, d\theta. \tag{3.22}$$

When n_c conductors are present, the total perturbation can be obtained as $\delta\psi = \sum_{i=1}^{n_c} \delta\psi_I$. Using the approach presented above, small changes to the magnetic field can be computed very fast by evaluation of the elliptic integrals (3.21) and (3.22) at positions (R, Z) once and substitution in (3.19). Afterwards, the fields resulting from the individual conductors only need to be rescaled to find $\delta\psi$ for other $\delta I_{\phi,i}$. In contrast to the Grad-Shafranov equation, a space discretization method is not needed here for the solution of $\delta\psi$.

Within this approximation, an explicit formulation to compute the magnetic flux ψ is now at hand. However, it is formulated in implicit form to define the magnetic state equations $c_{\mathrm{eq}}(\boldsymbol{\varphi}, q_{\mathrm{eq}}) = 0$, with the magnetic state q_{eq} corresponding here to the poloidal magnetic flux ψ,

$$c_{\mathrm{eq}} = \sum_i^{n_c} \frac{-\mu_0 \mathrm{R}\, \delta I_{\phi,i}}{k_i \pi} \sqrt{\frac{r_i'}{\mathrm{R}}} \left[\left(1 - \frac{k_i^2}{2} \right) K(k_i) - E(k_i) \right] + \psi_0 - \psi, \tag{3.23}$$

and ψ_0 being, again, the poloidal magnetic flux of the unperturbed equilibrium. As such, the adjoint derivations in chapter 5 include any (nonlinear or implicit) form of the magnetic state equations. The magnetic flux ψ will further be referred to as the magnetic state q_{eq} and the coil current changes $\delta I_{\phi,i}$ can be combined into the vector $\boldsymbol{\varphi}$, representing the control or design variables.

3.2 Plasma edge grid generation

As discussed in the introduction of this chapter, plasma edge transport models are typically discretized on a grid aligned to the magnetic field to avoid excessive numerical diffusion [27, 115, 118, 130, 135, 150]. To enable the use of structured grids, a coordinate transformation is typically used. Plasma edge grid generation is thus more than a spatial discretization process. A vector of continuous

state variables can therefore be attributed to this grid generator. Since an understanding of this continuous equivalent of the grid generator process is necessary when examining the options for its sensitivity calculation, this continuous meaning will be presented first. Afterwards, the discrete grid generation software that is developed within the context of this dissertation is described, with a focus on its automation. This requirement prohibits the straightforward use of existing plasma edge grid generator codes.

3.2.1 Continuous

Plasma edge transport equations are typically solved in a curvilinear poloidal coordinate system, that is aligned to the poloidal projection of the magnetic field. In figure 3.5 we sketch two mappings that are involved. The mapping

$$\boldsymbol{G_1}(\theta, r): \quad \mathbb{R}^2 \to \mathbb{R}^2, \quad \begin{pmatrix} r \\ \theta \end{pmatrix} \mapsto \begin{pmatrix} R(\theta, r) \\ Z(\theta, r) \end{pmatrix}$$

describes the mapping from the poloidal section (θ, r) of the curvilinear (θ, ϕ, r) coordinate system to the poloidal section (R, Z) of the basic cylindrical coordinate system $(\mathrm{R}, \phi, \mathrm{Z})$ [3, 46]. θ and r are respectively the coordinates along and perpendicular to the isolines of the poloidal flux, with the r coordinate directed outwards (see figure 3.5). The coordinate system (θ, ϕ, r) is again an orthogonal coordinate system. The transformation can be characterized by its metric coefficient matrix g_{ij}. In practice, since toroidal symmetry is assumed, only the metric coefficients $h_r = |\partial \boldsymbol{G_1}/\partial r|$, $h_\theta = |\partial \boldsymbol{G_1}/\partial \theta|$ and the Jacobian $J_1 = \sqrt{g} = \sqrt{\det[g_{ij}]}$ of (R, Z) with respect to (θ, r) are needed.

Due to the saddle point that the poloidal flux ψ exhibits at the X-point and due to the closed flux lines within the LCFS, so-called *cuts* need to be introduced to uniquely define the coordinate transformation. These cuts literally cut the core part at X-E'line and place it in the (θ, r)-frame between the inner and outer private flux zones, which are separated by the cut E-X. In the plasma edge solver, steps will be taken to impose the appropriate connections of the domains at these cuts.

Another important task of the grid generator is the choice of the plasma edge simulation domain. Traditionally, target surfaces were aligned with a grid

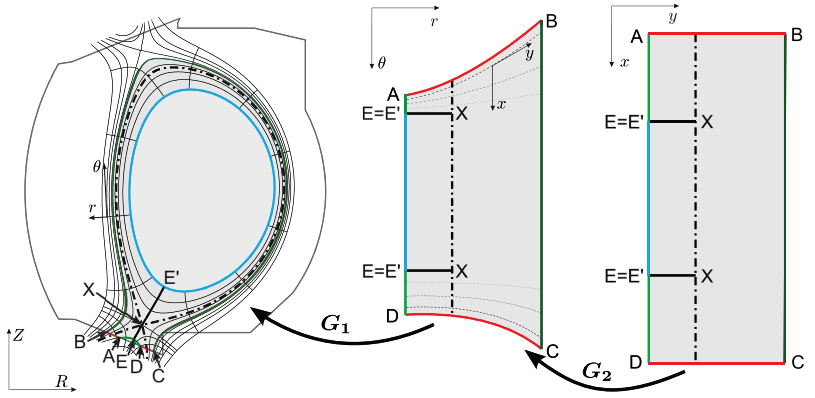


Figure 3.5: The grid generator mapping of the $(\mathrm{R}, \phi, \mathrm{Z})$ coordinate system, via the orthogonal (θ, ϕ, r) coordinate system, to the non-orthogonal (x, ϕ, y) coordinate system. Figure following Dekeyser [41].

boundary and the so-called outer wall boundary (B-C), core boundary (E'-E') and private flux boundary (A-E-D) have always been aligned with a magnetic flux surface. The main reason was again the risk for excessive numerical diffusion. To incorporate the domain choice in the grid generator state variable, the plasma edge domain indicator function

$$\mathbb{1}_{\Omega_{\mathrm{pe}}}(\mathbf{r}) := \begin{cases} 1 & \text{if } \mathbf{r} \in \Omega_{\mathrm{pe}} \\ 0 & \text{if } \mathbf{r} \notin \Omega_{\mathrm{pe}} \end{cases} \tag{3.24}$$

is introduced.

Although the poloidal curvilinear (θ, ϕ, r) coordinate system is convenient for exploiting the toroidal symmetry of the simulations, the transport equations are best expressed in a parallel-diamagnetic-radial $(\parallel, \perp, r)$ coordinate system, because it separates best the various transport phenomena and their time-scales. This is an orthogonal curvilinear coordinate system in which the $\mathbf{e}_\parallel$ axis is aligned to the magnetic field. The $\mathbf{e}_\perp$ and $\mathbf{e}_r$ coordinate axes are both perpendicular to the magnetic field vector $\boldsymbol{B}$. However, the diamagnetic axis lies in a surface of constant poloidal magnetic flux, while $\mathbf{e}_r$ is perpendicular to it. Since the momentum equation along the magnetic field lines will be projected onto the 2-D (θ, r) poloidal plane, a geometrical projection factor is still needed. To this end,

the poloidal pitch $b_\theta = B_\theta/\|\boldsymbol{B}\|$ is introduced, which is the cosine of the angle between the magnetic field $\boldsymbol{B}$ and its poloidal component $B_\theta\,\mathbf{e}_\theta$. The state vector of the grid generation can hence be defined as $\boldsymbol{q}_{\text{gg}} = \left(h_\theta, h_r, \sqrt{g}, b_\theta, \mathbb{1}_{\Omega_{\text{pe}}}\right)^T$ and the explicit grid generation procedure is symbolized as $\boldsymbol{q}_{\text{gg}} = \boldsymbol{c}_{\text{gg}}(q_{\text{eq}})$.

In an attempt to further represent the objectives of the grid generator in a continuous frame, one might introduce a second coordinate transformation that maps the plasma edge domain Ω_{pe} to a rectangular grid. Introducing the coordinate system (x, ϕ, y), the discretization process simply reduces to choosing appropriately spaced iso-x and y lines. This mapping

$$\boldsymbol{G_2}(x, y): \quad \mathbb{R}^2 \to \mathbb{R}^2, \quad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \theta(x, y) \\ r(x, y) \end{pmatrix}$$

then accommodates for the target geometry by relaxing the orthogonality constraint. The mapping $\boldsymbol{G}_2$ induces the transformation between the orthogonal coordinates system (θ, ϕ, r) and the non-orthogonal curvilinear (x, ϕ, y) coordinate system, where x is the coordinate along the isolines of poloidal magnetic flux lines (see a.o. [3, p. 88-89],[41, p. 71]). The x and y coordinates can in discrete sense be interpreted as the matrix coordinates of grid cells. It is clear that the freedom in this transformation entirely lies in the choice of the y-coordinate. Ideally, this choice maximizes the grid orthogonality, while smoothly adjusting for the solid geometry. Some algorithms directly pursuit this goal by using optimization algorithms [97]. Essentially, this is the most challenging part of the grid generator. It is understood that orthogonality in practice cannot be reached. In fact, orthogonality is often strongly abandoned near the target surface (see e.g. [126]). Therefore, it is essential to correctly account for this non-orthogonality in the discretization of plasma edge equations. Yet, up till present, non-orthogonality corrections are absent in most plasma edge solvers .

If one uses recent methods to impose boundary conditions for computational grids not aligned with the target structure [4, 27], the latter transformation becomes redundant, though the grid will not have a rectangular matrix form. Although the second coordinate transformation $\boldsymbol{G_2}$ can be seen as the continuous equivalent of the grid generator as presented in [15], it can equally be seen as a part of the discretization process. Indeed, with an appropriate discretization

that corrects for the non-orthogonality of grid cells, the mesh can be constructed directly in the orthogonal (θ, ϕ, r) coordinate system, then requiring the complete set of metric coefficients g_{ij} of the mapping $\boldsymbol{G_1} \circ \boldsymbol{G_2}$. It is mainly presented here to give insight into how the grid is mapped into a rectangular matrix form.

3.2.2 Discrete

In discrete form, the transformation $\boldsymbol{G_1}$ is dealt with by a plasma edge grid generator. Isolines of the poloidal magnetic flux ψ are traced using contour algorithms and each grid cell has two sides coinciding with two of these contours. Discrete metric coefficients are calculated from the cell geometries and represent cell widths in the direction of their coordinate axes. The Jacobian of the transformation is computed in the discrete grid generator as the cell volume.

High quality computational grids for solving the plasma edge transport equations are needed, in order to obtain accurate plasma edge profiles, e.g. to asses the target heat load. However, since such qualitative grids can only be achieved using quadrilateral grid cells aligned to the poloidal magnetic flux, every change in magnetic field induces a corresponding change in plasma edge grid. Therefore, the grid generator is a substantial part of the coupled simulation chain in figure 3.1 and must generate grids for the numerical evaluation of $\boldsymbol{q}_{\mathrm{gg}}(q_{\mathrm{eq}})$ in a completely automated fashion. The challenge is to combine grid and automation for each possible magnetic configuration.

In order to create these high quality grids, all spatial gradients should be sufficiently resolved. Given the typical plasma profiles, it is necessary to have a fine radial discretization towards the separatrix. Conversely, the poloidal resolution of the 2D grid can be rather crude near the midplane, which results in cells with a high aspect ratio. Grid cells are chosen here close to orthogonal for best numerical accuracy. Since the streamlines of $\nabla\psi$ are locally orthogonal to the poloidal flux per definition, they can be used to trace the radial grid lines [39]. This choice avoids twisted cell edges in the strongly elongated cells near the separatrix, as shown by the author in Ref. [11].

Further, the divertor region requires a strong poloidal grid refinement and a continuous adjustment of the nearly orthogonal grid at the midplane towards a

grid aligned with the target shape. In figure 3.6, a grid of the MATLAB grid generator is shown to illustrate these grid features. Here, no iterative procedure is used to enhance grid orthogonality in this region in contrast to most current plasma edge grid generators, such as CARRE [97]. Non-orthogonality in plasma edge computation will now be tackled with an appropriately enhanced discretization scheme [43]. The absence of this iterative procedure improves the predictability of the grid generation process, which facilitates the robust grid generation under different magnetic field inputs.

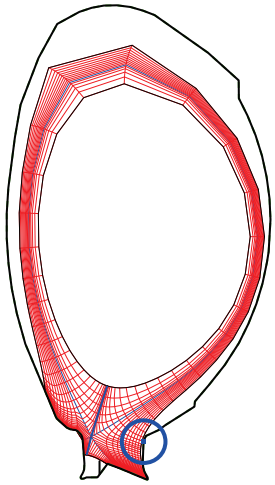


Figure 3.6: A computational grid for plasma edge transport simulations from the MATLAB grid generator. Remark the strong refinement towards target plates and separatrix. The blue circle highlights the point that determines the outer grid boundary (see text).

The automation of the grid generation needs some additional care. A first issue arises when detecting the X-points and the O-point (magnetic axis) of the magnetic configuration as this often requires additional, case specific, input. Indeed, X- and O-points can be detected by calculating the locations where $\nabla\psi = 0$. However, in general secondary X- and O-points may be present. The selection of the correct stationary point is done by first looking at the eigenvalues of the 2×2 curvature matrix at these locations. Their signs indicate whether the stationary

point is a saddle point or maximum of the flux function and hence a X- or O-point of the magnetic configuration, respectively. The primary X-point is then found by taking the X-point with the largest ψ-value within the vacuum vessel.

Secondly, the outer radial boundaries of the domain $\partial\Omega_{\text{pe}}$ should be selected in an appropriate way. The influence of the boundary conditions imposed here should be as small as possible as they are typically very crude (constant decay length boundary conditions for ion density and temperature and a recycling condition for neutrals, see section 3.3.3). Therefore, the simulated region is chosen to extend to the vessel wall as far as possible. In addition, the approximations for the objective functional, as will be explained in section 4.1, will benefit from these distant radial boundaries. However, until recently, these grid boundaries where restricted to coincide with magnetic flux surfaces that are fully contained, from target to target, in the vessel. For all these reasons, it is chosen to select the outermost magnetic flux surface in the SOL that does not cross any solid material in between the divertor targets. In figure 3.6, it can be seen that the outer grid boundary in the scrape-off layer (SOL) indeed nearly touches the vessel at the location highlighted with the blue circle.

In the private flux region, such a boundary can not always be found. Indeed, the outermost flux surface there might be a point. Since gridding up till this point would give a singularity in the grid cell sizes, an other approach is applied at this boundary. Either, a fictitious 'limiter' line is added that limits the radial extent of the simulated plasma in the private flux area. Alternatively, the private flux grid is chosen to cover a fixed flux fraction between the X-point and this singular point. Of course, a discretization scheme reaching all the way up to the reactor wall, as presented in Ref. [4] or [27], is a now natural way to approach this grid boundary problem. Given the additional complexity of the methods used in [4, 27], extending the optimization approach elaborated in this PhD thesis to this procedure is left for future work.

Finally, in order to verify robust behaviour of the grid generator, a test procedure with random inputs was implemented. This procedure generates a series of possible combinations for the controlled coil currents that obey the design constraints (discussed further in the next chapter). Using the perturbation approach,

the magnetic flux is computed and a grid generation is attempted. Grid generator errors and twisted cells are automatically detected and listed. Based on the results of these random input tests, implementations were generalized, and grid as well as constraint parameters were tuned to have the best overall grid performance. In this way, the chances that the simulations do not converge during the optimization cycle are minimized.

3.3 Plasma edge modelling

Most plasma edge models are a reduced form of the Braginskii [25] plasma fluid equations for electrons and ions, augmented with a model for neutral particles. For general notions on the Braginskii equations in a field aligned curvilinear coordinate system, the reader is referred to [3]. In this thesis, a simplified model is used in comparison to the more advanced hybrid fluid-kinetic B2-EIRENE code [115] to somewhat reduce computational time and facilitate the demonstration of the optimal design procedure. The edge plasma transport model of [5] is chosen, a slight variant of the model developed in the PhD thesis of Dekeyser [41]. The reader is referred to [41] for a more elaborate discussion of the model assumptions and implementation details.

3.3.1 Plasma model equations

Ion continuity equation

The model includes a single species plasma with ion mass m and charge state Z_i in a poloidal cross section of a toroidally symmetric tokamak for steady state. Charge neutrality is assumed, so that ion and electron densities n_i and n_e are related by $n_\mathrm{e} = Z_\mathrm{i} n_\mathrm{i}$. The ion density n_i is governed by the ion continuity equation

$$\frac{\partial n_\mathrm{i}}{\partial t} + \nabla \cdot (n_\mathrm{i} \mathbf{V}_\mathrm{i}) = S_{n_\mathrm{i}}. \tag{3.25}$$

The particle source

$$S_{n_\mathrm{i}} = n_\mathrm{e} n_\mathrm{n} K_\mathrm{i} - n_\mathrm{i} n_\mathrm{e} K_\mathrm{r}$$

includes a source term due to ionization of neutral particles, with n_n the neutral density, and a sink that accounts for volume recombination of ions and electrons to neutrals. K_i and K_r are the rate coefficients for these ionization and volume recombination interactions, respectively.

Ion parallel momentum equation

Due to the transport anisotropy, the ion velocity $\mathbf{V}_\mathrm{i}$ is modelled differently in the parallel and radial direction. The parallel velocity $u_\parallel$ is modelled by the parallel momentum equation

$$\frac{\partial m n_\mathrm{i} u_\parallel}{\partial t} + \nabla \cdot \left(m n_\mathrm{i} u_\parallel \mathbf{V}_\mathrm{i} - \eta^\mathrm{i} \nabla u_\parallel \right) = S_{m u_\parallel} - \nabla_\parallel p, \tag{3.26}$$

with $p = (1 + Z_\mathrm{i})\, n_\mathrm{i} T$ the plasma pressure and T a combined ion-electron-neutral temperature $T = T_\mathrm{i} = T_\mathrm{e} = T_\mathrm{n}$, $\eta^\mathrm{i} = diag(\eta^\mathrm{i}_\theta, \eta^\mathrm{i}_r)$ the (anisotropic) ion viscosity tensor, and where $\nabla_\parallel = \mathbf{e}_\parallel \cdot \nabla$. This equation results from adding the parallel momentum equations for ions and electrons and neglecting electron inertia because of the small electron mass ($m_\mathrm{e} \ll m$). Furthermore, electron viscosity is small ($\eta^e \sim m_\mathrm{e}^{1/2}$) and thus neglected. Following the classic B2 code, several curvature terms are neglected as well as viscosity tensor contributions including radial or diagmagnetic velocities [3]. It should be noted that several of these terms are included in more recent versions of the code [121]. The parallel momentum source $S_{m u_\parallel}$ includes momentum loss due to volume recombination and momentum loss due to charge-exchange collisions with neutral particles. The parallel momentum source is thus given by

$$S_{m u_\parallel} = m n_\mathrm{e} n_\mathrm{n} K_\mathrm{i} u_{\mathrm{n}\parallel} - m n_\mathrm{i} n_\mathrm{e} K_\mathrm{r} u_\parallel - m n_\mathrm{i} n_\mathrm{n} K_\mathrm{cx} \left(u_\parallel - u_{\mathrm{n}\parallel} \right),$$

with K_cx the rate coefficient for charge-exchange collisions.

In the radial direction, the velocity u_r is determined by the so-called *anomalous* turbulent transport, which is at current not completely understood. Therefore, radial transport is described by an empirical anomalous diffusion relation

$$n_\mathrm{i} u_r = -D^\mathrm{i} \nabla_r n_\mathrm{i},$$

where a spatially constant anomalous diffusion coefficient D^i will be considered in the work. This relation can be used to eliminate the radial ion velocity u_r from the plasma continuity equation and the energy equation, introduced below.

Neutral pressure diffusion equation

The neutral flow is modelled using the pressure diffusion equation

$$\frac{\partial n_{\mathrm{n}}}{\partial t} - \nabla \cdot \left(D_p^{\mathrm{n}} \nabla p_{\mathrm{n}} \right) = S_{n_{\mathrm{n}}}, \tag{3.27}$$

which is solved for the neutral pressure $p_{\mathrm{n}} = n_{\mathrm{n}} T$. Since neutrals are not confined to magnetic fields, D_p^{n} is an isotropic neutral pressure diffusion coefficient. The coefficient can be related to the ionization and charge exchange interactions by reformulating the static neutral momentum equation, where a balance between pressure gradient force and momentum source terms is kept [41]. The coefficient is then given by

$$D_p^{\mathrm{n}} = \frac{1}{m \left(n_{\mathrm{i}} K_{\mathrm{cx}} + n_{\mathrm{e}} K_{\mathrm{i}} \right)}. \tag{3.28}$$

Similar expressions can be found in a.o. [116]. The neutral sources $S_{n_{\mathrm{n}}}$ due to the neutral-plasma interactions are given by $S_{n_{\mathrm{n}}} = -S_{n_{\mathrm{i}}}$. As the neutral velocities resulting from this simple pressure-diffusion model tend to be unrealistically high, the neutral velocity $u_{\mathrm{n}\parallel}$ is set to zero in the ion momentum source. The validity of this assumption and the neutral model itself might be questioned, and should be dealt with when moving from methodological development to true design studies.

Total internal energy equation

For the energy transport, it is assumed that ions, electrons and neutrals are in perfect equilibrium. Denoting the density of species a (i = ions, e = electrons or n = neutrals) by n_a, the energy transport can thus be modelled by an internal energy equation

$$\frac{\partial}{\partial t} \left(\frac{3}{2} \sum_a n_a \right) + \nabla \cdot \left(\frac{5}{2} \sum_{a=i,e} \mathbf{\Gamma}^a T - \kappa \nabla T \right) = S_E, \tag{3.29}$$

solved for the equal-assumed species temperature $T \equiv T_{\mathrm{e}} = T_{\mathrm{i}} = T_{\mathrm{n}}$ of ions, electrons, and neutrals. The particle flux vectors $\mathbf{\Gamma}^a$ are given by $\mathbf{\Gamma}^{\mathrm{i}} = n_{\mathrm{i}} \mathbf{V}_{\mathrm{i}}$ and $\mathbf{\Gamma}^{\mathrm{e}} = Z_{\mathrm{i}} \mathbf{\Gamma}^{\mathrm{i}}$ for ions and electrons respectively. Again, the energy convection due to neutral flux $\mathbf{\Gamma}^{\mathrm{n}} = n_{\mathrm{n}} \mathbf{V}_{\mathrm{n}}$ is neglected in the current model because of the over-estimated values of $u_{\mathrm{n}\parallel}$. The heat conductivity tensor $\kappa = \sum \kappa^a = diag(\kappa_\theta, \kappa_r)$

includes contributions from ions, electrons and neutrals. The plasma heat conduction is again anisotropic, while the neutral contribution is isotropic, and thus $\kappa_\theta = \kappa_\theta^{\mathrm{e}} + \kappa_\theta^{\mathrm{i}} + \kappa^{\mathrm{n}}$ and $\kappa_r = \kappa_r^{\mathrm{e}} + \kappa_r^{\mathrm{i}} + \kappa^{\mathrm{n}}$. The included energy sources S_E are energy losses through carbon radiation and energy losses related to ionization events. These sources are given by

$$S_E = -E_{\mathrm{i}} n_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} - c_z n_{\mathrm{i}} n_{\mathrm{e}} L_z,$$

with a spatially constant impurity fraction c_z, a radiative loss function $L_z(T)$ and an energy loss E_{i} per ionization event. In order to simplify notation, the heat flux $\mathbf{Q}$ is defined as

$$\mathbf{Q} = \frac{5}{2} \sum_{a=i,e} \mathbf{\Gamma}^a T - \kappa \nabla T,$$

with components

$$Q_\theta = \frac{5}{2} \sum_{a=i,e} \Gamma_\theta^a T - \kappa_\theta \nabla_\theta T,$$
$$Q_r = \frac{5}{2} \sum_{a=i,e} \Gamma_r^a T - \kappa_r \nabla_r T.$$

Rate coefficients, radiative loss function and transport coefficients

The same set of analytical expressions as in [41] is used to approximate the strongly temperature-dependent rate coefficients for electron impact ionization, radiative recombination, and for charge exchange, as well as for the radiative loss function L_z. In the parallel direction, plasma transport coefficients are set according to Braginskii [25]. Also these coefficients have a highly nonlinear dependence on temperature. The poloidal transport coefficients are obtained by projecting these parallel coefficients onto the poloidal plane and neglecting contributions from diamagnetic transport:

$$\eta_\theta^{\mathrm{i}} = \frac{4}{3} b_\theta^2 \eta_0^{\mathrm{i}},$$
$$\kappa_\theta^{\mathrm{e}} = b_\theta^2 \kappa_\parallel^{\mathrm{e}},$$
$$\kappa_\theta^{\mathrm{i}} = b_\theta^2 \kappa_\parallel^{\mathrm{i}}.$$

The parallel viscosity $\eta_{\parallel}^{\mathrm{i}} = 4/3\,\eta_0^{\mathrm{i}}$ is obtained here from elaborating the contribution of the first order viscosity tensor to the parallel momentum equation (see e.g. Ref. [3, p. 41]).

The expression $\kappa^{\mathrm{n}} = \chi^{\mathrm{n}} p_{\mathrm{n}} D_p^{\mathrm{n}}$ is used for the neutral conductivity, with the value $\chi^{\mathrm{n}} = 0.2$ used further in this thesis. The radial plasma transport coefficients are given by

$$\begin{aligned}
\eta_r^{\mathrm{i}} &= \nu^{\mathrm{i}} m n_{\mathrm{i}}, \\
\kappa_r^{\mathrm{e}} &= \chi^{\mathrm{e}} Z_{\mathrm{i}} n_{\mathrm{i}}, \\
\kappa_r^{\mathrm{i}} &= \chi^{\mathrm{i}} n_{\mathrm{i}},
\end{aligned}$$

with $\nu^{\mathrm{i}}, \chi^{\mathrm{e}}$ and χ^{i} coefficients calibrated from experiments or turbulence models.

Remarks on the model choice

It should be noted that alternative formulations of the neutral fluid model and its interaction with the edge plasma hold several potential improvements to this model, such as e.g. the recent improvements to the neutral pressure diffusion equation by Dekeyser [39]. Ultimately, more accurate neutral models as simulated with Monte Carlo codes (e.g. EIRENE) are desirable. Similarly, the introduction of a multispecies plasma edge model might improve the accuracy of the radiation calculation. Yet, the plasma edge model is deliberately kept as simple as possible while including main features to provide first results and the basic principles of the new design approach. Although the presented plasma edge model is simplified compared to these codes, it has shown to reproduce some important features, such as low and high recycling, as well as roll-over to detachment [39]. Improved modelling issues are planned to be addressed in future work.

3.3.2 Plasma model equations in convection-diffusion form

It is convenient for further derivations to introduce the vector of plasma state variables $\boldsymbol{q}_{\mathrm{pe}} = (n_{\mathrm{i}}, u_{\parallel}, p_{\mathrm{n}}, T)^T$. The steady-state plasma edge transport equations

can then be written succinctly as a set of coupled convection-diffusion equations $\mathcal{B}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = 0$, with

$$\begin{aligned}\mathcal{B}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = & S(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) \\ & - \frac{1}{\sqrt{g}} \frac{\partial}{\partial \theta} \left(\frac{\sqrt{g}}{h_\theta} C^\theta(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) - \frac{\sqrt{g}}{h_\theta^2} D^\theta(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) \frac{\partial \boldsymbol{q}_{\mathrm{pe}}}{\partial \theta} \right) \\ & - \frac{1}{\sqrt{g}} \frac{\partial}{\partial r} \left(\frac{\sqrt{g}}{h_r} C^r(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) - \frac{\sqrt{g}}{h_r^2} D^r(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) \frac{\partial \boldsymbol{q}_{\mathrm{pe}}}{\partial r} \right).\end{aligned} \tag{3.30}$$

Here, the vectors

$$C^\theta(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = \begin{pmatrix} n_{\mathrm{i}} u_\theta \\ m n_{\mathrm{i}} u_\theta u_\parallel \\ 0 \\ \frac{5}{2}(1 + Z_{\mathrm{i}}) n_{\mathrm{i}} u_\theta T \end{pmatrix} \qquad \text{and} \qquad C^r(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

represent the poloidal and radial convective flux, and $u_\theta = b_\theta u_\parallel$ the ion poloidal velocity. $D^\theta(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = \mathrm{diag}(0, \eta_\theta^{\mathrm{i}}, D_p^{\mathrm{n}}, \kappa_\theta)$ and

$$D^r(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = \begin{pmatrix} D^{\mathrm{i}} & 0 & 0 & 0 \\ m D^{\mathrm{i}} u_\parallel & \eta_r^{\mathrm{i}} & 0 & 0 \\ 0 & 0 & D_p^{\mathrm{n}} & 0 \\ \frac{5}{2}(1 + Z_{\mathrm{i}}) D^{\mathrm{i}} T & 0 & 0 & \kappa_r \end{pmatrix}$$

are matrices containing respectively the poloidal and radial diffusion coefficients. The sources are combined in the vector[†]

$$S(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}, \nabla_\theta \boldsymbol{q}_{\mathrm{pe}}, \nabla_r \boldsymbol{q}_{\mathrm{pe}}) = \begin{pmatrix} n_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} - n_{\mathrm{i}} n_{\mathrm{e}} K_{\mathrm{r}} \\ -\frac{b_\theta}{h_\theta} \frac{\partial p}{\partial \theta} - m n_{\mathrm{i}} n_{\mathrm{e}} K_{\mathrm{r}} u_\parallel - m n_{\mathrm{i}} n_{\mathrm{n}} K_{\mathrm{cx}} u_\parallel \\ n_{\mathrm{i}} n_{\mathrm{e}} K_{\mathrm{r}} - n_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} \\ -E_{\mathrm{i}} n_{\mathrm{e}} n_{\mathrm{n}} K_{\mathrm{i}} - c_z n_{\mathrm{i}} n_{\mathrm{e}} L_z \end{pmatrix}.$$

Using this form of the plasma edge transport equations will facilitate a comprehensive derivation of the adjoint equations.

[†]The arguments $\nabla_\theta \boldsymbol{q}_{\mathrm{pe}}$ and $\nabla_r \boldsymbol{q}_{\mathrm{pe}}$ are considered separately for further derivations. In contrast to the source in the complete model of Dekeyser [41], a radial gradient of $\boldsymbol{q}_{\mathrm{pe}}$ does not occur here. Nevertheless, the argument is kept for generality.

3.3.3 Boundary conditions

To obtain a solution, the model equations $\mathcal{B} = 0$ are combined with appropriate boundary conditions, written formally as $\mathcal{C} = 0$. Together, they form the plasma edge state equations $\boldsymbol{c}_{\mathrm{pe}} = \left(\mathcal{B}^{\top}, \mathcal{C}^{\top}\right)^{\top} = 0$. For convenience of the reader, the different grid boundaries are illustrated in figure 3.7. The figure illustrates the boundaries in the lay-out of the computational grid, that corresponds to the $x-y$ coordinate system of figure 3.5).

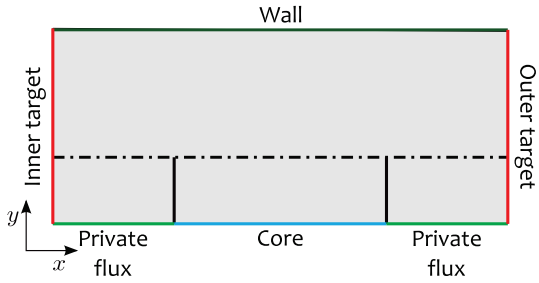


Figure 3.7: The plasma edge grid in the matrix layout ($x - y$ coordinates in figure 3.5) and its boundaries.

Target boundaries

At the solid targets, the ions and electrons of the plasma recombine to neutrals. A plasma sink is thus formed, that accelerates the plasma towards the targets. Due to the different masses and sound speeds of ions and electrons, electrons will accelerate more than ions. Whereas the largest share of the plasma is quasi-neutral, a thin electrostatic sheath will thus be formed near the targets. At the entrance of this sheath, the Bohm criterium imposes that the parallel velocity $u_{\parallel}$ reaches isothermal sound speed $c_{\mathrm{s}} = \left((T_{\mathrm{i}} + T_{\mathrm{e}})/m\right)^{1/2} = \left(2T/m\right)^{1/2}$ [134]. Since the plasma flow is assumed ambipolar ($\mathbf{V}_{\mathrm{e}} = \mathbf{V}_{\mathrm{i}}$) in the model, these boundary conditions are imposed at the sheath entrance. However, the width of this sheath is characterized by the Debye length and is only in the order of $10^{-5}\,\mathrm{m}$. The parallel velocity is therefore conveniently set to sound speed at the

target boundary itself as [41]

$$\boldsymbol{\Gamma}^{\mathrm{i}} \cdot \boldsymbol{\nu} = \boldsymbol{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu}, \qquad \text{with} \qquad \boldsymbol{\Gamma}_{\mathrm{t}} \equiv n_{\mathrm{i}} u_{\theta} \mathbf{e}_{\theta} = \pm n_{\mathrm{i}} c_{\mathrm{s}} b_{\theta} \mathbf{e}_{\theta}$$

with $\boldsymbol{\nu}$ the unit vector perpendicular to the domain boundary and pointing outwards of the domain Ω_{pe}, and where the sign of $\boldsymbol{\Gamma}_{\mathrm{t}}$ is chosen to have outflow at the considered boundary. A fraction R_c of the ions is recycled in the plasma as neutrals. Hence, the boundary condition for neutrals is $\boldsymbol{\Gamma}^{\mathrm{n}} \cdot \boldsymbol{\nu} = -R_c \boldsymbol{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu}$. The boundary condition for the energy equation is finally given by

$$\mathbf{Q} \cdot \boldsymbol{\nu} = \delta_{\mathrm{sh}} T \boldsymbol{\Gamma}_{\mathrm{t}} \cdot \boldsymbol{\nu} \equiv \mathbf{Q}_{\mathrm{t}} \cdot \boldsymbol{\nu}.$$

The energy outflow is thus proportional to the total sheath transmission factor $\delta_{\mathrm{sh}} = \delta^i_{sh} + \delta^e_{sh} + \delta^{pot}_{sh} - \alpha R_c \left(\delta^i_{sh} + \delta^{pot}_{sh}\right)$ [39], with δ^i_{sh} and δ^e_{sh} ion and electron sheath transmission coefficients, respectively, and δ^{pot}_{sh} the contribution of the sheath potential to electron heat. Typical values are $\delta^i_{sh} \approx 2.5$, $\delta^e_{sh} \approx 2$, and $\delta^{pot}_{sh} \approx 3.1$, but depends on the underlying assumptions [3, 133]. The remaining contribution $\alpha R_c \left(\delta^i_{sh} + \delta^{pot}_{sh}\right)$ represents the energy recycled by the neutrals from the ions, with $\alpha = 0.5$ a fixed fraction of energy that is recycled by the neutrals. The total sheath transmission factor then becomes $\delta_{sh} \approx 4.8$. In summary, the target boundary conditions can be represented by the vector

$$0 = \mathfrak{C}_t = \begin{pmatrix} \left(-\boldsymbol{\Gamma}_{\mathrm{t}} + \boldsymbol{\Gamma}^{\mathrm{i}}\right) \cdot \boldsymbol{\nu} \\ \pm c_{\mathrm{s}} - u_{\parallel} \\ \left(R_c \boldsymbol{\Gamma}_{\mathrm{t}} + \boldsymbol{\Gamma}^{\mathrm{n}}\right) \cdot \boldsymbol{\nu} \\ \left(-\mathbf{Q}_{\mathrm{t}} + \mathbf{Q}\right) \cdot \boldsymbol{\nu} \end{pmatrix}. \tag{3.31}$$

Core boundary

The core boundary is typically a surface of constant magnetic flux ψ, on which the plasma properties are more or less constant. Therefore, only boundary conditions for the radial direction need to be imposed. For the ion continuity equation, either the density n_{c} or the flux can be specified. Similarly, the temperature or the energy flux Q_{c} can be imposed for the internal energy equation. Since the objective is optimal design of the magnetic configuration, these boundary conditions should be chosen to keep the operation point of the tokamak constant throughout the design process. As such, the density and the total energy flux

leaving the core are imposed, two typical operation parameters. For the momentum equation, the parallel velocity is assumed to be zero at the core boundary. For neutrals, one might assume that all neutral particles ionize before they ever reach the core boundary, and therefore $\Gamma_r^{\mathrm{n}} = 0$. However, for smaller machines, this is often not true. Since the realistic study of small present day machines is aimed for in this PhD thesis, a leakage condition proportional to the product of local neutral density and neutral thermal speed $c_{\mathrm{n}} = \left(8T/(\pi m)\right)^{1/2}$ used here [33], $\Gamma_r^{\mathrm{n}} = -\alpha_{\mathrm{c}} c_{\mathrm{n}} n_{\mathrm{n}}$. In short, the core boundary conditions can be jointly represented as

$$0 = \mathfrak{C}_c = \begin{pmatrix} n_{\mathrm{c}} - n_{\mathrm{i}} \\ -u_{\parallel} \\ -\alpha_{\mathrm{c}} c_{\mathrm{n}} n_{\mathrm{n}} - \Gamma_r^{\mathrm{n}} \\ Q_{\mathrm{c}} - Q_r \end{pmatrix}. \tag{3.32}$$

Wall and private flux boundaries

The wall and private flux boundaries are treated similarly. Although recently methods have been developed to calculate up to the physical wall [4, 27], a flux surface is chosen to approximate the boundary in most simulations. Typically exponential decay of density and temperature is imposed with prescribed decay lengths λ_n and λ_T, and the radial component of the parallel velocity is assumed zero. When a Monte Carlo code is used to solve the kinetic equations for these neutrals, they are typically reflected at the physical boundary. Since a fluid neutral model is used, the particles are assumed to recycle at the domain boundaries. Furthermore, the effect of the pump is incorporated by adding an outflow term $\Gamma_{\mathrm{p}} = \alpha_{\mathrm{p}} c_{\mathrm{n}} n_{\mathrm{n}}$, where the absorption coefficient α_{p} can be obtained from the volumetric pumping speed $L\,[\mathrm{m}^3/\mathrm{s}]$ according to [114]

$$L = 36.38\, A\, \alpha_{\mathrm{p}} \sqrt{\frac{T}{m}}, \tag{3.33}$$

with A the surface over which particles are pumped in the model, T the particle temperature in degrees Kelvin and m the particle mass in [a.m.u.]. The boundary

conditions then read

$$0 = \mathfrak{C}_{p,w} = \begin{pmatrix} -D^{\mathrm{i}}\, n_{\mathrm{i}}/\lambda_n - \nu_r D^{\mathrm{i}} \nabla_r n_{\mathrm{i}} \\ -\nu_r \eta_r^{\mathrm{i}} \nabla_r u_{\parallel} \\ -\alpha_{\mathrm{p}} c_{\mathrm{n}} n_{\mathrm{n}} + \nu_r R_c \Gamma_r^{\mathrm{i}} + \nu_r \Gamma_r^{\mathrm{n}} \\ -\kappa_r\, T/\lambda_T - \nu_r \kappa_r \nabla_r T \end{pmatrix}, \tag{3.34}$$

where $\nu_r = \boldsymbol{\nu} \cdot \mathbf{e}_r$.

3.3.4 Discretizing and solving the plasma edge equations

The plasma edge transport equations are solved using a Finite Volume Method implementation by Dekeyser [41] that resembles the fluid part of more elaborate plasma edge code packages, such as the B2-EIRENE [115] code. A detailed treatment of the B2 code is found in [3]. Apart from the model simplifications, a main feature of Dekeyser's MATLAB implementation is the introduction of a 9-point stencil that corrects for the non-orthogonality of the grid cells [40, 41]. In B2, these non-orthogonalities are thus far neglected. Especially for the isotropic neutrals this is an important improvement.

Like in B2, the transport equations are solved using a segregated solver, in which a pressure-correction equation for incompressible flows is used to simultaneously solve the continuity equation for a density correction and velocity updates [3]. Pseudo-time stepping is used to iterate the equations from a chosen initial guess until a steady-state solution is found.

Solving a correction equation

The individual equations are solved using a correction equation. To illustrate the iterative solver, consider one of the time-dependent equations in a general convection-diffusion form,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{C}\phi - D\nabla\phi) = S, \tag{3.35}$$

with $\boldsymbol{C} = \left(C^{\theta}, C^{r}\right)^{\top}$ and $D = \mathrm{diag}(D^{\theta}, D^{r})$ the convective and diffusive coefficients of the equation, ϕ the unknown plasma state variable, and S the (nonlinear)

source terms. Starting from a given initial source S_0 and plasma state variable ϕ_0, the code solves for linearized updates $\Delta\phi$ from the implicit correction equation

$$\frac{\partial \Delta\phi}{\partial t} + \nabla \cdot \left(\left(\boldsymbol{C} + \frac{\partial \boldsymbol{C}}{\partial \phi}\phi \right) \Delta\phi - D\nabla(\Delta\phi) \right) - S_\mathrm{v}\Delta\phi = \mathcal{R}, \tag{3.36}$$

where the residual $\mathcal{R}$ is defined through

$$\mathcal{R} = S - \nabla \cdot (\boldsymbol{C}\phi - D\nabla\phi) \tag{3.37}$$

and where the source S is decomposed in a constant S_c and a variable $S_\mathrm{v}\phi$ contribution. Remark that this is an approximate linearization, since the highly nonlinear transport coefficients in D are not linearized.

Boundary conditions

The boundary conditions are imposed using guard cells. As illustrated in figure 3.8, these cells are chosen to have a very small width so that only the flux across the boundary will have a significant value. The boundary flux F can then be imposed through the choice of source in the guard cell, since [†]

$$F \equiv \boldsymbol{\Gamma} \cdot \boldsymbol{\nu}\Sigma \approx S\Omega = (S_\mathrm{c} + S_\mathrm{v}\phi)\Omega,$$

with Σ the surface of the guard cell's face and Ω the volume of the guard cell. More difficult is imposing Dirichlet conditions like $\phi = \phi_0$. To impose such conditions, a numerical trick is used. Through multiplying the sources using a very big number $\mathbb{B}$, the flux across the boundary becomes negligible and hence

$$S = S_\mathrm{c} + S_\mathrm{v}\phi \approx 0. \tag{3.38}$$

The Dirichlet condition can thus be imposed by choosing $S_\mathrm{c} = -\mathbb{B}\phi_0$ and $S_\mathrm{v} = \mathbb{B}$. Similarly, Neumann and more complex mixed conditions can be imposed by choosing appropriate source contributions [3].

[†]The vector $\boldsymbol{\nu}$ refers in this expression to the outwards pointing guard cell normal, as indicated in figure 3.8.

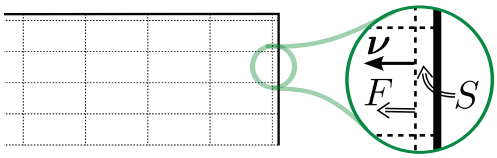


Figure 3.8: Infinitesimal guard cells are used to impose boundary conditions. Figure based on Refs. [3, 41].

3.4 Conclusions

As a first step towards optimal magnetic divertor design, the plasma edge transport solver developed by Dekeyser [41] is integrated with a magnetic field solver and a grid generation approach to automatically evaluate the effect of changes to the control coils on the target heat load. More precisely, a combination of three sequential code parts computes the target heat load from the control currents $\boldsymbol{\varphi}$. First, the magnetic flux function has to be evaluated, represented by the magnetic field equations $c_{\mathrm{eq}}\left(\boldsymbol{\varphi}, q_{\mathrm{eq}}\right) = 0$. For this evaluation, a perturbation approach has been presented that starts from an initial magnetic equilibrium and uses a vacuum approximation for magnetic field changes. Then, a plasma edge grid is generated that is fully aligned to this magnetic field to avoid excessive numerical diffusion. For this part, a new plasma edge grid generator is developed that complies with the automation requirements. As such, the grid generator state variables are automatically obtained from the function $\boldsymbol{q}_{\mathrm{gg}} = \boldsymbol{c}_{\mathrm{gg}}\left(q_{\mathrm{eq}}\right)$. In the last step, the set of plasma edge transport equations $\boldsymbol{c}_{\mathrm{pe}}\left(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}\right) = 0$ is solved for the plasma state $\boldsymbol{q}_{\mathrm{pe}}$ (and thus the target heat load) using a segregated solver. These plasma edge equations consist of ion continuity and momentum equation, a pressure diffusion equation for neutral transport and a total internal energy equation, with highly nonlinear sources and transport coefficients. The steady-state solution is obtained through a slowly converging segregated solver. The presented model now facilitates the study of optimization methods for magnetic divertor design. This is the topic of the next chapter.

4

Automated Design of a JET Configuration

After introducing the basic principles of PDE-constrained optimization in chapter 2 and after introducing the model in chapter 3, the path towards magnetic divertor design has been cleared. Nevertheless, the design problem remains to be converted into a numerical optimization problem. This is a crucial exercise, which involves defining how designs should be compared, to what extent parameters can be controlled and which designs are desirable or impossible. To this end, each of the components in the constrained mathematical optimization formulation (2.1) should be defined. Including the model equations as defined in the previous chapter, this optimization problem reads

$$\begin{aligned} \min_{\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}} \quad & \mathcal{J}\left(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}\right) \\ s.t. \quad & c_{\mathrm{eq}}(\boldsymbol{\varphi}, q_{\mathrm{eq}}) = 0, \\ & \boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}}) - \boldsymbol{q}_{\mathrm{gg}} = 0, \\ & \boldsymbol{c}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = 0, \\ & \boldsymbol{h}(\boldsymbol{\varphi}, q_{\mathrm{eq}}) \leqslant 0, \end{aligned} \tag{4.1}$$

where the control vector $\boldsymbol{\varphi}$ includes the coil currents $I_{\phi,i}$ and with q_{eq}, $\boldsymbol{q}_{\mathrm{gg}}$, and $\boldsymbol{q}_{\mathrm{pe}}$ the magnetic field, grid generator, and plasma state variables as defined in chapter 3. The equations $c_{\mathrm{eq}}(\boldsymbol{\varphi}, q_{\mathrm{eq}}) = 0$ and $\boldsymbol{c}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = 0$ reflect the implicit form of magnetic field and plasma edge equations, while the explicit expression

$\boldsymbol{q}_{\mathrm{gg}} = \boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}})$ reflects the explicit form of the grid generation process. Notice that the admissible set of design variables Φ_{ad} is accounted for explicitly through the (inequality) design constraints $\boldsymbol{h}(\boldsymbol{\varphi}, q_{\mathrm{eq}}) \leqslant 0$. Again, the model equations can be used to formally eliminate the state variables and obtain the reduced cost function as $\hat{\mathcal{J}}(\boldsymbol{\varphi}) := \mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}(\boldsymbol{\varphi}), \boldsymbol{q}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi})), \boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi}))))$.

In this thesis, the design focus is on spreading the target heat flux as homogeneously as possible, by means of magnetic field changes only. However, the optimization framework is very well suited for finding compromises between several design goals and constraints, such as simultaneously improving the pumping capability of the configuration as well. Likewise, other design variables might be included, such as the target shape parametrization of Ref. [37]. This chapter first discusses the appropriate formulation of the design objective $\mathcal{J}$ in section 4.1. Afterwards, the introduction of design constraints $\boldsymbol{h}$ is considered in section 4.2. These constraints should avoid that undesirable magnetic configurations are considered or that the nominal coil currents are exceeded.

After defining all terms mathematically, a solution to the optimization problem is sought with a nested optimization approach. Although the basics of nested optimization have been discussed in chapter 2, design constraints remain to be incorporated. For this purpose, a nonlinear gradient projection method is chosen in section 4.3 to assure optimization robustness. The constrained optimization approach itself is subsequently introduced in section 4.4. Finally, the entire approach is illustrated using a sample case with typical Joint European Torus (JET) [119] parameters in section 4.5.[†]

4.1 A suitable objective functional formulation

Following Dekeyser [39], the objective functional representing the level of heat spreading can be expressed by

$$\mathcal{J}_0\left(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}\right) = \frac{1}{2}\int_{S_t} \left(Q_{\mathrm{o}} - Q_{\mathrm{d}}\right)^2 d\boldsymbol{\sigma}, \tag{4.2}$$

[†]Parts of this chapter have been published in "Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N. & Reiter, D. (2014). An Automated Approach to Magnetic Divertor Configuration Design. *Nuclear Fusion*, **55**".

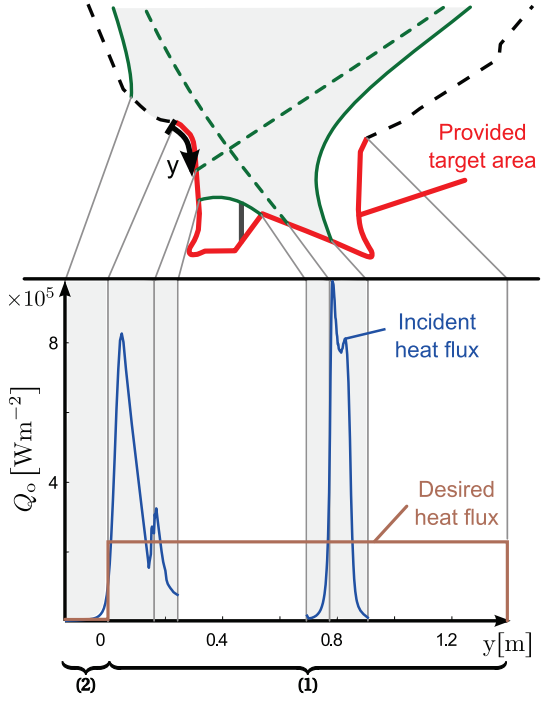


Figure 4.1: Computation of the objective functional. Outer grid boundaries (solid green lines), gridded area (grey area), and predefined target area (red) are illustrated in a close-up of the divertor region (top). Difference between the desired (brown) and initial (not optimized, blue) heat flux along the projected target area coordinate 'y' (bottom). (1) and (2) correspond to the surface over which is integrated for terms (1) and (2) of Equation (4.4), respectively.

where the integral is taken over the target surface S_t and $\boldsymbol{\nu}$ represents the outward pointing unit normal. This objective aims at bringing the heat flux density perpendicular to the target surface $Q_\mathrm{o} = \mathbf{Q}_\mathrm{o} \cdot \boldsymbol{\nu}$ as close as possible to a desirable spatially constant heat flux profile Q_d. The subscript 'o' refers here to the heat load that is to be optimized. Indeed, the objective functional $\mathcal{J}_0\left(\boldsymbol{\varphi}, q_\mathrm{eq}, \boldsymbol{q}_\mathrm{gg}, \boldsymbol{q}_\mathrm{pe}\right)$ is minimal when the target area is loaded as uniformly as possible. In this case the best design relative to this particular choice of cost functional and plasma model is achieved. The computation of the objective functional is illustrated in

figure 4.1. The target area S_t is a fixed predefined part of the vessel chosen by the designer to serve this purpose and is typically covered with high-temperature erosion-resistant materials.

The target surface is subject to different heat load contributions. The first contribution is the heat $\mathbf{Q} \cdot \boldsymbol{\nu}$, transported to the target by conduction and convection of plasma and neutrals. According to the sheath conditions at the targets, this heat flux density equals $Q_\mathrm{t} = \mathbf{Q}_\mathrm{t} \cdot \boldsymbol{\nu} = \delta_\mathrm{sh} T \Gamma_\mathrm{t} \cdot \boldsymbol{\nu}$. Secondly, energy is released when ions and electrons recombine to neutrals at the target surface. The heat load due to this *surface recombination* amounts to $Q_\mathrm{sr} = \mathbf{Q}_\mathrm{sr} \cdot \boldsymbol{\nu} = E_\mathrm{p} \boldsymbol{\Gamma}_\mathrm{t} \cdot \boldsymbol{\nu}$, with E_p the potential energy of recombination, which equals 13.6 eV for Hydrogen. A third contribution comes from the ion kinetic energy, which is not included in the internal energy flux density $\mathbf{Q}$. However, since part of this kinetic energy is converted to neutral kinetic energy and the neutral kinetic energy is not modelled, both ion and neutral kinetic contributions to the heat load are neglected. Finally, also the contributions from electrical currents and radiation are neglected at this stage. The electrical currents are neglected since divertor currents are neglected throughout the model. Radiation is finally assumed to be distributed quite uniformly and therefore assumed not to affect the optimal heat load distribution. However, since radiation sources are often located in the divertor legs, one might consider calculating the heat load deposited by radiation using a Monte Carlo code, as done in Ref. [43]. Given these assumptions, the total target heat load is modelled in this thesis as

$$\begin{aligned} Q_\mathrm{o} &= Q_\mathrm{t} + Q_\mathrm{sr} \\ &= (\delta_\mathrm{sh} T + E_\mathrm{p}) n u_\theta \mathbf{e}_\theta \cdot \boldsymbol{\nu}. \end{aligned} \tag{4.3}$$

A practical problem in the evaluation of the objective functional is that not necessarily all heat load values for the target area are available from the code, as parts of the area may not be covered by the computational grid. It can be assumed however that in these regions, either far into the scrape-off layer (SOL) or private flux (PF) zone, heat fluxes are orders of magnitude lower than those near the separatrix. Moreover, as the grid extends to the flux surface that just touches solid material, the radial decay of heat flux towards PF zone or to the far SOL from this radial point on will be even steeper, due to shorter connection

length further out. Therefore, it is chosen here to neglect parallel target heat fluxes outside the numerical grid. It should be noted that these neglected zero heat fluxes do still contribute to the cost functional, as the integration of (4.2) is over the entire predefined target area, including the parts outside the numerical grid. This contribution stimulates better use of this part of the target area, e.g. by magnetic flux expansion. For a more precise treatment a discretization scheme reaching all the way up to the reactor wall, as elaborated in [4] or [27], should be considered.

As the first wall is not included into the objective functional (4.2), care has to be taken that the optimization procedure is not misled by a deflection of heat fluxes towards the first wall components to realize a decrease in cost functional. Indeed, this would irreparably damage the first wall. Therefore, a penalty term is added to the objective functional (term (2) in Equation (4.4) below). By setting the desired heat flux to the first wall $Q_{d,p}$ equal to zero, this term becomes zero when there is no parallel heat flux towards the first wall. To ensure convergence of the optimization problem, a Tikhonov regularization term is added (term (3) in Equation (4.4) below) [139, 140]. Notice that in this case, the regularization term also has economical relevance, since it avoids configurations with excessive Joule losses.

After adding these contributions to the main objective term $\mathfrak{J}_0$, the total objective functional becomes

$$\begin{aligned}\mathfrak{J}\left(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}\right) =& \lambda_Q \left(\underbrace{\mathfrak{J}_0}_{(1)} + \underbrace{\frac{1}{2}\int_{S_P} \left[\lambda_P \left(Q_{\mathrm{o}} - Q_{d,p}\right)\right]^2 d\sigma}_{(2)} \right) \\ &+ \underbrace{\frac{1}{2}\lambda_\varphi \sum_{i=1}^{n_{\boldsymbol{\varphi}}} I_{\phi,i}^2}_{(3)},\end{aligned} \tag{4.4}$$

where S_P is the area outside the predefined target area and λ_P, λ_Q, and λ_φ are weighting variables for the related terms. Additionally, λ_Q and λ_φ contain characteristic quantities for heat flux and current, respectively, to make the cost functional dimensionless and of the order of unity.

4.2 Introducing a core shape constraint

Given the objective functional of equation (4.4), the definition of additional constraints for the optimization problem is considered now. These constraints might firstly be introduced from a modelling perspective. Indeed, depending on the algorithms used for the three simulation blocks, the models might not be generally applicable for any $\boldsymbol{\varphi}$. Generalizing these blocks is very time-consuming and is often conflicting with our demand for acceptably fast simulations and optimization. A trade-off therefore arises. Either the models are improved for automated usage, or alternatively, additional constraints are introduced to define practical working limits. For example, a box constraint on the current can be applied to limit the errors made by the perturbation approach to magnetic field calculations presented in section 3.1.4. This means that an upper and lower limit, $\varphi_{i,\max}$ and $\varphi_{i,\min}$, are set for all currents φ_i:

$$\varphi_i \in \left[\varphi_{i,\min}, \varphi_{i,\max}\right]. \tag{4.5}$$

This can be easily included into the constraint vector $\boldsymbol{h}$ as $h_i = \varphi_{i,\min} - \varphi_i$ and $h_{i+n_{\boldsymbol{\varphi}}} = \varphi_i - \varphi_{i,\max}$. Likewise, such constraints can be tuned to achieve robust grid generation in the feasible set (set of allowed current combinations), e.g. avoiding multiple X-points. It should be noted that the latter constraint is at present mainly introduced to avoid additional complexity in plasma edge simulations.

A second reason to introduce constraints might be to incorporate design requirements not accounted for by the objective functional. For example, even for small magnetic field changes as will be considered in this work, the core might outgrow the vessel, causing a loss of confinement. This is of course undesirable from both a design as well as a robust simulation perspective. The resulting limiter configuration would not meet design requirements as the provision of sufficient neutral pumping possibilities or the guarantee of sufficient core purity. Therefore, additional (indirect) constraints on $\boldsymbol{\varphi}$ are put in place to prevent this phenomenon. A practical solution to guarantee confinement is constraining the plasma core spatially within a box, as illustrated in figure 4.2. Numerically, the

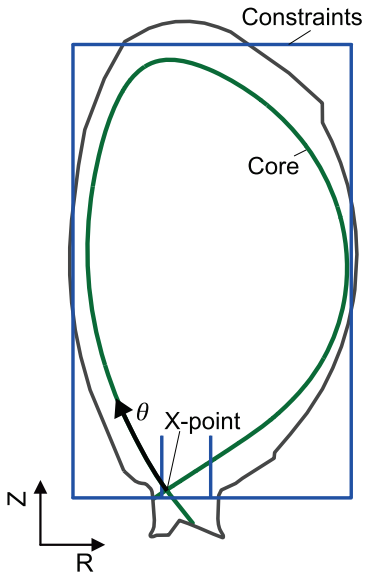


Figure 4.2: The constraints on the magnetic topology.

choice of constraints in figure 4.2 can be translated into inequality constraints:

$$\begin{aligned}
h_{1+2n_{\boldsymbol{\varphi}}} &= \max_{\theta\in\theta_{\text{core}}} \left(R_{\text{core}}(\theta)\right) - R_{\max} \leqslant 0, \\
h_{2+2n_{\boldsymbol{\varphi}}} &= \max_{\theta\in\theta_{\text{core}}} \left(Z_{\text{core}}(\theta)\right) - Z_{\max} \leqslant 0, \\
h_{3+2n_{\boldsymbol{\varphi}}} &= R_{\min} - \min_{\theta\in\theta_{\text{core}}} \left(R_{\text{core}}(\theta)\right) \leqslant 0, \\
h_{4+2n_{\boldsymbol{\varphi}}} &= Z_{\min,\text{X}} - Z_{\text{X}} \leqslant 0, \\
h_{5+2n_{\boldsymbol{\varphi}}} &= R_{\min,\text{X}} - R_{\text{X}} \leqslant 0 \qquad \text{and} \\
h_{6+2n_{\boldsymbol{\varphi}}} &= R_{\text{X}} - R_{\max,\text{X}} \leqslant 0,
\end{aligned} \tag{4.6}$$

with θ the coordinate along the last closed flux surface, $R_{\text{core}}(\theta)$ and $Z_{\text{core}}(\theta)$ the parametrization of the main plasma boundary, subscript X referring to the location of the X-point and $R_{\min}$, $R_{\max}$, $Z_{\max}$, $Z_{\min,\text{X}}$, $R_{\min,\text{X}}$ and $R_{\max,\text{X}}$ preset values for the constraint box. Convexity of the core shape is assumed so that there cannot be multiple outer core points and the gradient of these inequality constraints thus always exists. For later use, the reduced design constraints $\widehat{\boldsymbol{h}}(\boldsymbol{\varphi}) := \boldsymbol{h}(\boldsymbol{\varphi}, q_{\text{eq}}(\boldsymbol{\varphi}))$ are defined, of which the magnetic state variables are elim-

inated using the magnetic state equations $c_{\mathrm{eq}}(\boldsymbol{\varphi}, q_{\mathrm{eq}}) = 0$.

4.3 Projection onto the design constraints

As inequality constraints are in general only obeyed after convergence of the optimization problem, intermediate states of the optimization problem might demand evaluating objective functional values for which these constraints are not obeyed. This means that the core might be well outside the described box. Unfortunately grid generation and therefore plasma calculations and objective functional evaluations would be impossible in this case. The optimization procedure thus has to obey these constraints at any time.

This can be achieved in several ways. *Interior point methods*, for example, deal efficiently with many constraints by relaxing the complementarity conditions during optimization. However, given this specific optimization problem, where evaluations of $\hat{J}(\boldsymbol{\varphi})$ are much more expensive than evaluations of $\widehat{\boldsymbol{h}}(\boldsymbol{\varphi})$, a nonlinear *gradient projection method* is chosen [103]. Accordingly, the optimal design path is never compromised for feasibility gain, until a constraint boundary is exceeded. Then, the design variables are projected to the closest point in the admissible design space (in other words the point with the smallest change in currents), which does satisfy all design constraints. As the separatrix box constraint is strongly nonlinear, the projection $\boldsymbol{\varphi}^\star = \mathcal{P}(\boldsymbol{\varphi})$ of a point $\boldsymbol{\varphi}$ onto a point $\boldsymbol{\varphi}^\star$ that obeys all constraints is an optimization problem by itself and is given by

$$\mathcal{P}(\boldsymbol{\varphi}) = \arg\min_{\boldsymbol{\varphi}^\star \in \Phi_{ad}} ||\boldsymbol{\varphi}^\star - \boldsymbol{\varphi}||_2^2 \, s.t. \; \widehat{\boldsymbol{h}}(\boldsymbol{\varphi}^\star) \leqslant 0, \tag{4.7}$$

with $||\cdot||_2$ the Euclidean norm, and "arg min" the argument $\boldsymbol{\varphi}$ for which the function is minimal.

As this optimization problem can be as complex as the initial optimization problem, gradient projection methods are in general only used in combination with linear control constraints. However, in this case, the inequality constraints only depend on magnetic field variables and the magnetic field computation is many times faster than the plasma edge simulation. Therefore, the gradient projection method turns out to be appropriate here.

4.4 Solving the optimization problem

In order to solve the optimization problem (2.1), a (nested) Sequential Quadratic Programming (SQP) strategy with damped BFGS Hessian updating is chosen [103]. SQP was presented already in the context of one-shot optimization in chapter 2, where the state constrained optimization problem was linearized to obtain a linearly (equality) constrained quadratic subproblem in each iteration. In the current nested context, where SQP is applied to the reduced optimization problem

$$\begin{aligned} \min_{\boldsymbol{\varphi}} \quad & \hat{\mathfrak{J}}\left(\boldsymbol{\varphi}\right) \\ s.t. \quad & \widehat{\boldsymbol{h}}(\boldsymbol{\varphi}) \leqslant 0, \end{aligned} \tag{4.8}$$

SQP has a somewhat different meaning. Here, the SQP method aims at solving the inequality constrained optimization problem by solving a sequence of quadratic subproblems

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \nabla\hat{\mathfrak{J}}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T B_k \boldsymbol{p} \\ s.t. \quad & \widehat{\boldsymbol{h}}(\boldsymbol{\varphi}_k) + \nabla\widehat{\boldsymbol{h}}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} \leqslant 0, \end{aligned} \tag{4.9}$$

with k the optimization iteration index and $\boldsymbol{p} = \boldsymbol{\varphi}_{k+1} - \boldsymbol{\varphi}_k$ the step in control variables that is optimized in each iteration. B_k is the Hessian estimate from the damped BFGS approach presented in section 2.2.1. Remark that in presence of inequality constraints, the vector $\boldsymbol{y_k} = \nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_{k+1}) + \boldsymbol{\mu}^\top\nabla\boldsymbol{h}(\boldsymbol{\varphi}_{k+1}) - \nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_k) - \boldsymbol{\mu}^\top\nabla\boldsymbol{h}(\boldsymbol{\varphi}_k)$ should be used in the Hessian update rule of Eq. (2.27).

At this point, an adjoint based sensitivity calculation for evaluating the design sensitivity $\nabla\hat{\mathfrak{J}}$ of the integrated code is not yet readily available. As further discussed in chapter 5, especially the grid generator part hinders a full adjoint sensitivity calculation at this point. A solution will be offered in the very same chapter. For the time being, the gradients of the objective functional $\nabla\hat{\mathfrak{J}}$ and the constraint function $\nabla\widehat{\boldsymbol{h}}$ with respect to the control variables $\boldsymbol{\varphi}$ are therefore evaluated using finite difference calculations. In each direction $\boldsymbol{\delta\varphi}$ the directional sensitivity $\delta\hat{\mathfrak{J}}$ is then approximated with the (forward) finite difference evaluation

$$\delta\hat{\mathfrak{J}}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi}) \approx \Delta^{\varepsilon}\hat{\mathfrak{J}}(\boldsymbol{\varphi}) := \frac{\hat{\mathfrak{J}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}) - \hat{\mathfrak{J}}(\boldsymbol{\varphi})}{\varepsilon}, \tag{4.10}$$

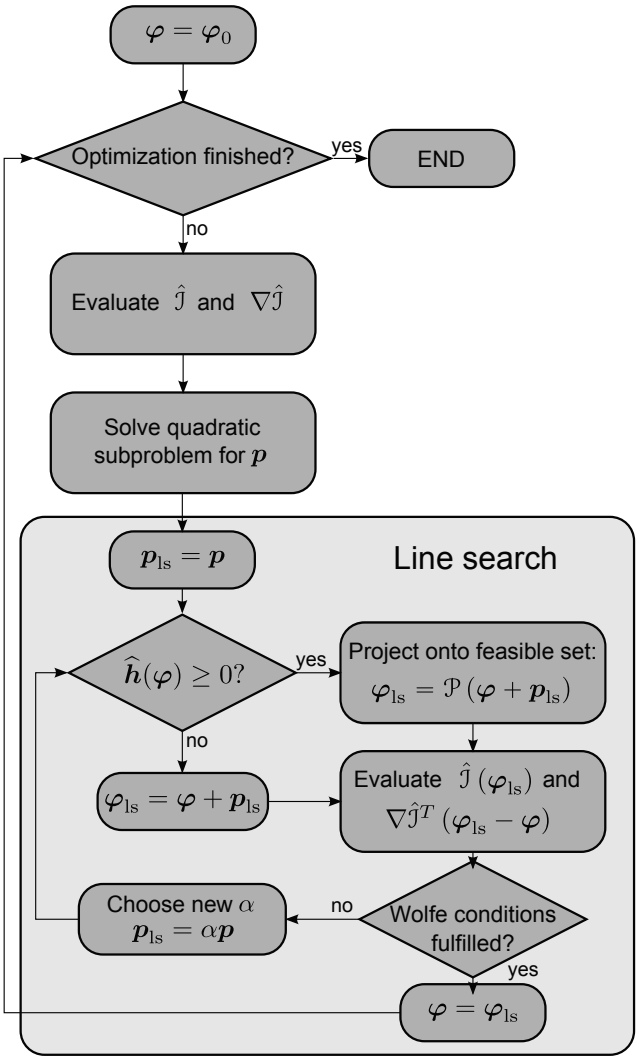


Figure 4.3: An overview of the implemented SQP optimization algorithm.

where the finite differentiation operation in direction $\boldsymbol{\delta\varphi}$ and with step size ε is denoted as $\Delta^{\varepsilon}(\cdot)$. Unfortunately, this means that for a control vector of length $n_{\boldsymbol{\varphi}}$, $n_{\boldsymbol{\varphi}} + 1$ cost function evaluations are needed, with a plasma edge transport

simulation for each evaluation. It should be noted that the evaluation of $\nabla\hat{\mathcal{J}}$ then requires many plasma edge simulations and is therefore computationally extremely demanding. Ultimately, more efficient adjoint sensitivity calculations of $\nabla\hat{\mathcal{J}}$ are targeted (cfr. chapter 5). Computing the constraint gradient $\nabla\widehat{\boldsymbol{h}}$, on the other hand, is relatively cheap as it involves magnetic field calculations only. Additionally, an adjoint approach to sensitivity calculation of $\nabla\widehat{\boldsymbol{h}}$ would not be of great benefit. Indeed, the number of adjoint magnetic field simulations scales with the number of nonlinear inequality constraints in the vector $\widehat{\boldsymbol{h}}$. Therefore, an adjoint sensitivity calculation approach should only be considered in case a significant number of controls $n_{\boldsymbol{\varphi}}$ is used.

The finite difference step ε should be chosen to balance truncation and cancellation errors. The former errors are associated to the validity of ignoring the nonlinear terms in the Taylor expansion of $\hat{\mathcal{J}}(\boldsymbol{\varphi}+\varepsilon\boldsymbol{\delta\varphi})$. Therefore, they increase when the step size ε increases. The latter errors are due to the digital representation of numbers with a finite amount of digits and thus increase with decreasing finite difference step. All numbers smaller than machine precision ε_{m} are therefore neglected. Since squaring operations are involved in the objective functional calculation, one should especially make sure the step size is not too small. Indeed, for every step smaller than $\varepsilon=\varepsilon_{\mathrm{m}}^{1/2}$, the cancellation error might lead to a total loss of significant digits in the sensitivity[†]. Therefore, the finite difference step size is chosen as $\varepsilon=\varepsilon_{\mathrm{m}}^{1/4}$.

Once the quadratic subproblem (4.9) is solved, an inexact line search is performed along the obtained direction $\boldsymbol{p}$. This comprises searching a step size that sufficiently decreases the cost functional. To this end, the Wolfe criteria, equations (2.36) and (2.37), are used in the algorithm. When the search line crosses constraints, the projection algorithm explained in section 4.3 is applied to bend the search line along the constraint surface. In this way, the search direction can become slightly altered compared to the initial search direction. Remark that the general formulation of the Wolfe conditions in section 2.3 allows for this.

[†]Notice that for the step size ε to make sense, $\boldsymbol{\delta\varphi}$ should have the same order of magnitude as $\boldsymbol{\varphi}$. In the code the design variables are simply normalized with a characteristic current magnitude such that unit vectors can be used for $\boldsymbol{\delta\varphi}$.

The art of a good line search algorithm is finding a step length that obeys the Wolfe conditions with as few cost functional evaluations as possible. Advanced interpolation strategies can be applied that use former cost functional and gradient information to search the lowest value along the line. In our algorithm a line search strategy developed by Moré and Thuente [99] is adapted to account for the projection of infeasible control variable combinations onto the feasible set. A schematic overview of the implemented SQP procedure is given in figure 4.3.

4.5 Automated design procedure applied to a JET configuration

4.5.1 Test case specification

In this section the novel automated design method is, in a first feasibility study, applied for a JET reference configuration obtained from earlier numerical edge transport studies [146]. The results of this study have been published in Ref. [12]. Model parameters (detailed below) are chosen to keep the edge transport model as simple as possible (low recycling conditions) to focus the attention on the capability of the magnetic optimization aspects in this first illustrative application. Yet, realistic parameters are used to obtain a relevant test case. The initial magnetic field is an EFIT reconstruction of an experimental L-mode JET shot 80966 named HT3L from the c28b JET campaign (2011). It is characterized by a horizontal outer target, a high triangularity ($\delta = 0.38$) and $I_p/B_T = 0.8\ \mathrm{MA\,T^{-1}}$. A poloidal cut of the magnetic configuration is displayed in figure 4.4. The electric currents that will be optimized are those carried by four fictitious divertor coils. The number of conductors is set to four to keep the computational costs for gradient computations manageable. These conductors, as well as the target area used in the cost functional formulation (4.4), are illustrated in the same figure. Apart from the box constraints on the design currents and the core shape constraints, a linear constraint

$$\varphi_{\mathrm{minsum}} \leqslant \sum_{i=1}^{n_{\boldsymbol{\varphi}}} \varphi_i \leqslant \varphi_{\mathrm{maxsum}} \tag{4.11}$$

is introduced to limit the change to the more distant core magnetic field and the according loss of grid quality and magnetic model validity. That is, if the currents in negative and positive $\mathbf{e}_\phi$-direction approximately add to zero, the far magnetic field will be small. The constraint parameters are listed in table 4.1 for completeness.

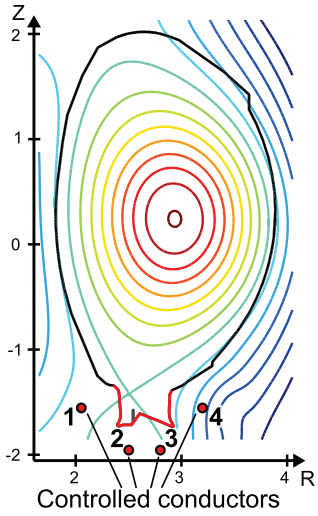


Figure 4.4: The set-up of the JET-case for magnetic field design. Four new conductors (numbered) are placed near the predefined target area (red). The contour lines represent magnetic flux surfaces.

The plasma edge simulation is performed on a 240×80 grid. This is finer than most plasma edge simulations to accommodate for the fluid neutral transport. Transport and boundary condition parameters are set according to the values in table 4.2. Remark that the temperature decay length λ_T is large in comparison to models with a separate energy equation for the neutrals. This accounts for the dominant contribution of the neutrals to heat transport in the far SOL [33]. All ions reaching wall and target domain boundaries are recycled as neutrals. At the private flux the same recycling condition is kept, but additionally neutrals are pumped here. As the area of the last private flux surface

	position [m]	min. $\boldsymbol{\varphi}$ [kA]	max. $\boldsymbol{\varphi}$ [kA]
φ_1	$(2.05, -1.55)$	-97.1	97.1
φ_2	$(2.50, -1.95)$	-97.1	97.1
φ_3	$(2.80, -1.95)$	-97.1	97.1
φ_4	$(3.20, -1.55)$	-97.1	97.1
$\sum_{i=1}^{n_{\boldsymbol{\varphi}}} \varphi_i$		-161.8	161.8
$R_{\min}$	1.83		
$Z_{\max}$	1.85		
$R_{\max}$	3.85		
$Z_{\min,\mathrm{X}}$	-1.44		
$R_{\min,\mathrm{X}}$	2.475		
$R_{\max,\mathrm{X}}$	2.83		

Table 4.1: The values of the different parameters that appear in the constraint expressions (4.5),(4.6),(4.11).

constant	value	units
D^{i}	0.8	$\mathrm{m^2 s^{-1}}$
χ^{i}	1	$\mathrm{m^2 s^{-1}}$
χ^{e}	1	$\mathrm{m^2 s^{-1}}$
χ^{n}	0.2	$\mathrm{m^2 s^{-1}}$
ν^{i}	0.8	$\mathrm{m^2 s^{-1}}$
λ_n	0.05	-
λ_T	1	-
n_{c}	$1.3 \cdot 10^{19}$	$\mathrm{m^{-3}}$
Q_{c}	3	MW
L	7.34	$\mathrm{m^3 s^{-1}}$
α_{c}	0.2	-

Table 4.2: The values of transport and boundary condition parameters used for the JET case.

is varied during simulations, we keep the effective pumping fixed for an assumed constant boundary temperature. The absorption coefficient α_{p} is then found from

equation (3.33),

$$L = 36.38\, A\, \alpha_{\mathrm{p}} \sqrt{\frac{T}{m}},$$

where $L = 7.34$ m^{3}s^{-1} at room temperature is maintained in all simulations. Our choice of model parameters results in low recycling divertor conditions, as a nearly constant temperature profile along flux surfaces in the divertor region and a linear relation between target particle flux and upstream density were observed.

4.5.2 Results

Two magnetic field optimization applications will be discussed in this section. They only differ in the weighting of the wall heating penalty term (term (2) of Equation (4.4)), which is either $\lambda_P = 1$ or $\lambda_P = 10$. The weighting of this wall heating term reflects to what extent one wants to tolerate a higher value for main objective term (1) of Equation (4.4) to avoid the need for target area expansion. If the magnetic configuration of an existing reactor is designed, one would rather try to avoid changing the material configuration at the wall. In this case, the λ_P parameter can be given a higher value. In case of a new reactor design, a lower λ_P will explore the opportunity to enlarge the divertor target area. The other objective functional weighting variables are $\lambda_Q = 2.85 \cdot 10^{-12}$ m^{2}W^{-2} and $\lambda_\varphi = 1 \cdot 10^{-12}$ A^{-2}.

Optimization with low wall heating penalty

In figure 4.5, the change of control variables and cost functional are given for the different optimization stages in case of low wall heating penalty. Starting from zero perturbation currents, negative and positive currents are introduced by the optimization loop to reduce the cost functional value. In figure 4.5b, the different contributions of equation 4.4 are presented. Although the total cost functional is decreasing, not every single term in equation 4.4 is. One can observe that a compromise is found between decreasing the heat flux to the targets and limiting the increase in wall heating penalty and current penalty.

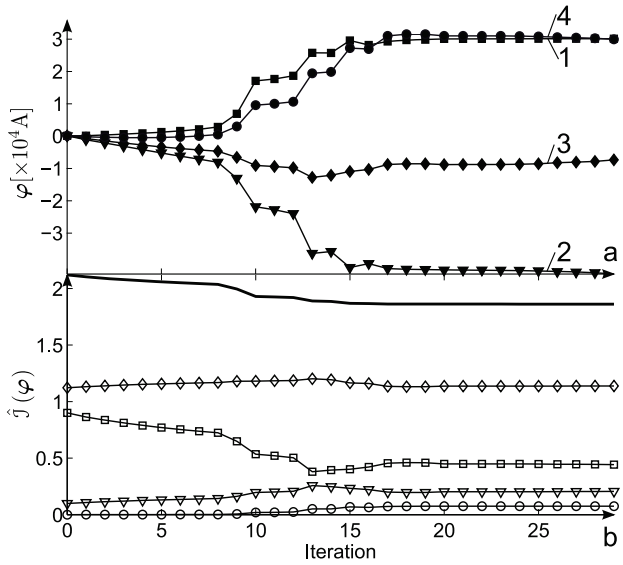


Figure 4.5: a) The change of control current magnitudes (numbers of coils corresponding to fig. 4.4) and **b)** the change of the different cost functional contributions as a function of the number of optimization iterations performed for $\lambda_P = 1$. —: total cost functional, -◇-and -□-: respectively not simulated and simulated part of term (1) in Equation (4.4), -▽-: wall penalty (term (2) in Equation (4.4)), -○-: control (current) penalty (term (3) in Equation (4.4))[†].

The dominant reduction in cost functional is achieved by decreasing the main objective term for heat spreading over the target surface. To verify this, the simulated target heat flux is displayed in figure 4.6 at different optimization stages. Indeed, a sharp reduction of the heat peaks is observed at the cost of a tolerable heat flux increase outside the defined target area, as the increased heat peak

[†]This figure was first published "Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N. & Reiter, D. (2014). An Automated Approach to Magnetic Divertor Configuration Design. *Nuclear Fusion*, **55**". Because of an inconsistency in the code in the usage of the poloidal flux per radian ψ and total poloidal flux $\Psi = 2\pi\psi$, the currents $\boldsymbol{\varphi}$ were overestimated by a factor 2π. This is resolved by rescaling all currents. The other results in the paper remain correct.

around $y = 0$ remains well below the other two peaks. The maximal heat load has decreased by 36%.

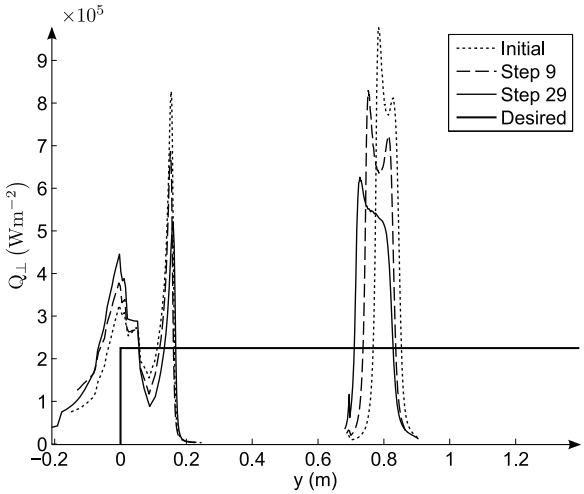


Figure 4.6: The change in heat loading from initial (···) to optimized (—) magnetic configuration for $\lambda_P = 1$.

In figure 4.7, the initial and optimized magnetic fields are shown. It can be observed that the opposing current pairs 1, 2 and 3, 4 (Fig. 4.5) cause an expansion of the field lines at the target area, spreading the heat flux analogously to the X-divertor concept [85]. It should be noted, however, that this configuration is by no means an optimization in general, in particular not for standard JET operation conditions (high recycling or detached divertor states). The results rather demonstrate that the automated procedure developed here provides plausible trends relative to an underlying physical edge model. Although it is logical to maximize flux expansion in this low recycling test case, an automated design result for high recycling or even detached conditions might be harder to interpret and is currently outside the validity range of our still simplified underlying plasma model. Finally, it should be noted that the X-point is constrained to its lower boundary $Z_{\min,X}$ (Table 4.1).

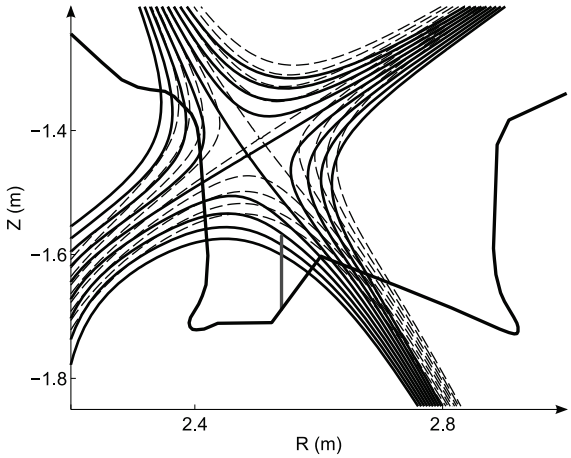


Figure 4.7: The change in poloidal magnetic flux from initial (– –) to the optimized (—) configuration for $\lambda_P = 1$.

Optimization with high wall heating penalty

Although the optimization procedure provides an interesting decrease in maximal heat flux using a low wall heating penalty of $\lambda_P = 1$, the 37% heat load increase outside the target area would require changing some first wall tiles for target material. In an existing reactor, this may be undesirable. Therefore, it is demonstrated here how increasing the penalty factor λ_P can account for this. Thus, λ_P is increased by a factor 10 to make the first wall heating penalty more dominant in the objective functional compared to the other terms and the optimization is repeated. Indeed, according to figure 4.8b, the main reduction in cost functional is achieved by removing all wall heat load contributions at the inner target in the first optimization step (-▽- curve). It is only afterwards that the target heat term is reduced (-□- curve). In figure 4.9 the reduction of this wall heating can be verified. In order to achieve this result, the X-point was moved away from the inner target area boundary by the optimization procedure (Figure 4.10), at the cost of a narrowing of the flux tubes and a corresponding rise in the heat peaks. When relaxing this design requirement one obviously retrieves again a heat peak

reduction, as could be seen for the optimization with $\lambda_P = 1$.

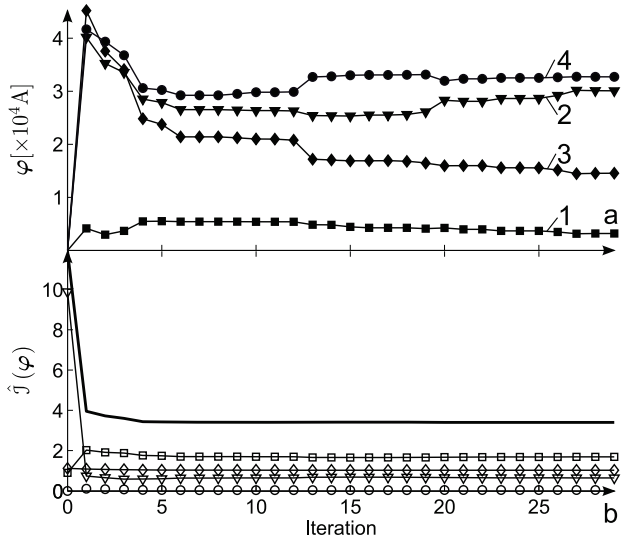


Figure 4.8: a) The change of control current magnitudes (numbers of coils corresponding to fig. 4.4) and **b)** the change of the different cost functional contributions as a function of the number of optimization iterations performed for $\lambda_P = 10$. —: total cost functional, –◇–and –□–: respectively not simulated and simulated part of term (1) in Equation (4.4), –▽–: wall penalty (term (2) in Equation (4.4)), –○–: control (current) penalty (term (3) in Equation (4.4))†.

4.6 Conclusion

An optimization-based approach for automated magnetic field design of tokamak divertors has been presented. To this end, an objective functional for magnetic divertor design is defined. Special care is taken to ensure design requirements not accounted for in the objective functional, e.g. core purity and neutral pumping,

†See footnote of figure 4.5.

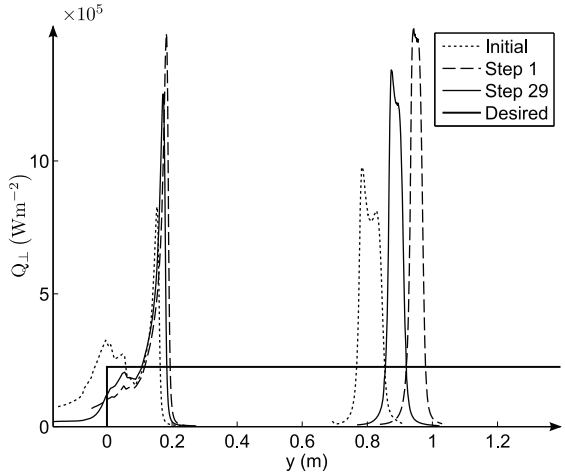


Figure 4.9: The change in heat loading from initial (···) to optimized (—) magnetic configuration for $\lambda_P = 10$.

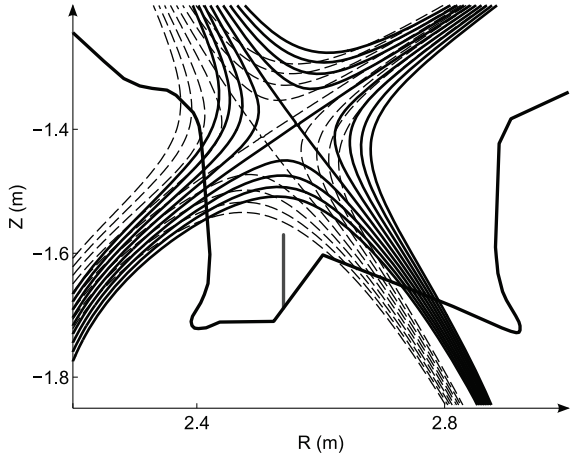


Figure 4.10: The change in poloidal magnetic flux from initial (−−) to the optimized (—) configuration for $\lambda_P = 10$.

are accounted for in appropriate optimization constraints that prevent limiter configurations. Then, an SQP solver was proposed to solve the inequality constrained optimization problem. Finite difference sensitivities of cost functional with respect to the coil currents were calculated, using the integrated modelling code presented in chapter 3. A nonlinear gradient projection approach was used to successfully robustify the optimization procedure against potential code crashes.

The optimal design approach has finally been demonstrated on a realistic JET magnetic field as a test case. The results for this reduced plasma edge model seem to indicate that the overall automated optimization loop provides plausible trends, and a significant sensitivity of edge plasma flow solutions with respect to magnetic configuration details. Additionally, it has been shown how the choice of objective functional reflects the design requirements. To this end, the influence of different weightings for cost functional terms, representing the relative importance of different design criteria, is investigated for a JET configuration.

Although currently reduced models are used for magnetic field as well as for plasma edge computations, other models can be integrated into the approach too. Unfortunately, plasma edge simulations are still computationally very costly, even for the reduced plasma model. The gradient evaluation cost could however greatly be reduced by using the adjoint approach for gradient computation, discussed in chapter 2. Using this approach, the time needed for a gradient calculation becomes in principle independent of the number of design variables. Hence, adjoint methods would allow for more design freedom and more complex plasma models. Therefore, the adjoint methods for magnetic field design are explored in the next chapter.

5

In-Parts Adjoint Sensitivities for Efficient Magnetic Divertor Design

In section 2.1.2, the adjoint approach to sensitivity calculation was introduced. The most important advantage of this method is, that it provides an affordable gradient calculation for scalar valued objective functions, independent of the number of design variables. Given that the plasma edge transport simulation presented in section 3.3 easily takes several hours on a single workstation, adjoint methods provide a reasonable efficiency gain already at a small number of control variables. If one wishes to include a more elaborated multispecies transport code such as B2-EIRENE, CPU-time of a single simulation sharply increases and might range up to a year (paralleled over a cluster) for simulation of power plant relevant conditions [41]. It is clear that for design or sensitivity studies with the latter code adjoint gradients are indispensable. Whether one aims for a combined design of divertor shape and magnetic configuration or a single sensitivity study of the target heat load with respect to a couple of coils, it is needless to say that both would largely benefit from adjoint sensitivity calculation.

Yet, as will be discussed in section 5.1, the direct application of the adjoint approach to the model chain of chapter 3 is rather cumbersome. Therefore, a more practical semi-discrete in-parts adjoint approach to sensitivity calculation is presented. This approach aims at using the continuous adjoint framework for the

plasma edge model that had been established by W. Dekeyser at KU Leuven [41], while circumventing excessive derivational efforts. The approach is conceptually presented in section 5.2. After deriving the continuous adjoint equations and their boundary equations in section 5.3, the method is further elaborated in section 5.4. There, the effects of discretization and boundary condition implementation within this semi-discrete approach are considered in detail. In section 5.5, the correctness of the approach is verified using a grid sensitivity study. Finally, in section 5.6, the advantages of this method are illustrated for magnetic configuration design.[†]

5.1 Motivation

For the optimization problem (4.1) the adjoint approach as introduced in section 2.1.2 can be directly applied if the model constraints and state variables are combined into the vector functions $\boldsymbol{c} = (c_{\mathrm{eq}}, \boldsymbol{c}_{\mathrm{gg}}{}^{\top} - \boldsymbol{q}_{\mathrm{gg}}^{\top}, \boldsymbol{c}_{\mathrm{pe}}{}^{\top})^{\top}$ and $\boldsymbol{q} = (q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}^{\top}, \boldsymbol{q}_{\mathrm{pe}}^{\top})^{\top}$, respectively. Remember that the grid generator state variables $\boldsymbol{q}_{\mathrm{gg}}$ are supposed to be given by an explicit function call $\boldsymbol{q}_{\mathrm{gg}} = \boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}})$, which explains the different shape of this component of $\boldsymbol{c}$. The adjoint variable vector function $\boldsymbol{q}^*$ can then analogously be decomposed in its components $\boldsymbol{q}^* = (q_{\mathrm{eq}}^*, \boldsymbol{q}_{\mathrm{gg}}^{*\top}, \boldsymbol{q}_{\mathrm{pe}}^{*\top})^{\top}$. Substituting these components of $\boldsymbol{c}$, $\boldsymbol{q}$ and $\boldsymbol{q}^*$ in equation (2.17) reveals the separate adjoint model equations [‡]

$$\begin{aligned}
\partial_{\boldsymbol{q}_{\mathrm{pe}}} \boldsymbol{c}_{\mathrm{pe}}{}^{*}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}})\, \boldsymbol{q}_{\mathrm{pe}}^* &= -\nabla_{\boldsymbol{q}_{\mathrm{pe}}} \mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}), \\
-\boldsymbol{q}_{\mathrm{gg}}^* &= -\nabla_{\boldsymbol{q}_{\mathrm{gg}}} \mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) - \partial_{\boldsymbol{q}_{\mathrm{gg}}} \boldsymbol{c}_{\mathrm{pe}}{}^{*}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}})\, \boldsymbol{q}_{\mathrm{pe}}^*, \\
\partial_{q_{\mathrm{eq}}} c_{\mathrm{eq}}^*(\boldsymbol{\varphi}, q_{\mathrm{eq}})\, q_{\mathrm{eq}}^* &= -\nabla_{q_{\mathrm{eq}}} \mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) - \partial_{q_{\mathrm{eq}}} \boldsymbol{c}_{\mathrm{gg}}{}^{*}(q_{\mathrm{eq}})\, \boldsymbol{q}_{\mathrm{gg}}^*,
\end{aligned} \tag{5.1}$$

which are solved for the adjoint variables $\boldsymbol{q}_{\mathrm{pe}}^*$, $\boldsymbol{q}_{\mathrm{gg}}^*$, and q_{eq}^*. Notice that the adjoint equations, like the state equations, can be performed sequentially, but in reversed

[†]Parts of this chapter have been published in "Blommaert, M., Heumann, H., Baelmans, M., Gauger, N.R. & Reiter, D. (2016). Towards Automated Magnetic Divertor Design for Optimal Heat Exhaust. *ESAIM: Proceedings and Surveys*, **53**, 49–63" and "Blommaert, M., Baelmans, M., Dekeyser, W., Gauger, N. & Reiter, D. (2015). A novel approach to magnetic divertor configuration design. *Journal of Nuclear Materials*, **463**, 1220–1224".

[‡]Notice that the mathematical liberty is taken here to suppose that the function $\mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}})$ can be analogously written with a vector argument $\boldsymbol{q}$ as $\mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}) \equiv \mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}})$.

order. In sequence, the system of equations (5.1) displays the adjoint plasma edge equations, the adjoint grid generator, and the adjoint magnetic field equations. The right hand sides of these equations contains the linearized objective function source terms. In the last two equations, this source term exists of two parts: the direct linearization of the source term with respect to the state variable of that equation and the indirect contribution through the dependences $\boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}})$ and $\boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}}(q_{\mathrm{eq}}))$, respectively. After solving the state and adjoint equations, the design sensitivity (2.20) reduces to

$$\nabla\hat{\mathcal{J}} = \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) + \partial_{\boldsymbol{\varphi}} c^*_{\mathrm{eq}}(\boldsymbol{\varphi}, q_{\mathrm{eq}})\, q^*_{\mathrm{eq}}. \tag{5.2}$$

For this fully continuous adjoint gradient calculation, the linearizations of the model equations $\boldsymbol{c}_{\mathrm{pe}}$, $\boldsymbol{c}_{\mathrm{gg}}$, c_{eq} and the objective functional $\mathcal{J}$ with respect to the control variables $\boldsymbol{\varphi}$ and the state variables q_{eq}, $\boldsymbol{q}_{\mathrm{gg}}$, and $\boldsymbol{q}_{\mathrm{pe}}$ are thus needed. Notice that the continuous adjoint plasma edge equations $\partial_{\boldsymbol{q}_{\mathrm{pe}}}\boldsymbol{c}_{\mathrm{pe}}{}^*\boldsymbol{q}^*_{\mathrm{pe}}$ and their implementation were dealt with by Dekeyser [41] and will be presented in section 5.3. The perturbation magnetic field model, presented in chapter 3, easily reveals its derivatives as well. Also for the more elaborate magnetic field models, e.g. based on the free boundary equilibrium equations, the derivatives could be obtained with some effort. The discrete adjoint derivatives of the FBE simulation software presented in chapter 6 are even readily available, since an exact Newton solver is used [72]. Therefore, only the partial derivative of the plasma edge transport equations with respect to the grid variables $\partial_{\boldsymbol{q}_{\mathrm{gg}}}\boldsymbol{c}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}})$ and the grid generator derivative $\partial_{q_{\mathrm{eq}}}\boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}})$ are still needed.

For several reasons, obtaining these remaining derivatives is cumbersome:

- Deriving an expression for $\partial_{\boldsymbol{q}_{\mathrm{gg}}}\boldsymbol{c}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}})$ is error-prone, since the metric coefficients and geometric projection factors in $\boldsymbol{q}_{\mathrm{gg}}$ are present in all terms of the equations $\boldsymbol{c}_{\mathrm{pe}}$ directly, and indirectly via e.g. transport coefficients. Additionally, the dependence of the plasma edge transport equations on the changing domain boundary Ω_{pe} should be quantified [†].

[†]The derivative $\partial_{\boldsymbol{q}_{\mathrm{gg}}}\boldsymbol{c}_{\mathrm{pe}}$ includes a derivative with respect to the indicator function $\partial_{\boldsymbol{q}_{\mathrm{gg}}}\mathbb{1}_{\Omega_{\mathrm{pe}}}$. Note that in theory this derivative is only defined in a weak sense. These *topological* derivatives are thoroughly discussed in literature (see e.g. Ref. [131]). Since this expression is only

- A continuous adjoint approach independent of the plasma grid and discretization is difficult, since the domain boundaries are a part of the grid generator state. Additionally, in chapter 6, the magnetic field model will be improved by integrating a free boundary equilibrium solver. The equilibrium equations are solved in this code using a first order finite element solver. Since Hadamard formulations for the sensitivities of the grid boundary movement require high smoothness of the magnetic state solution [67], they are not compatible with this first order finite element solver. The latter eliminates the option of reformulating the sensitivities based on boundary data only to cancel out the mesh dependence.
- Also a linearization of the discrete grid generator is difficult, since the grid generation code is not easily expressed in a closed equation. Additionally, it is a complex routine containing several functions with hidden source and many if-clauses. As such, also AD-tools are not easily applicable.

Luckily, the very low amount of computational time needed to solve the magnetic equilibrium and perform the grid generation in comparison to the plasma edge transport simulation triggers the use of a solely in-parts adjoint approach. It is the hope that this approach might combine the advantages of the adjoint approach for computational time gain on the plasma edge simulation with a more practical sensitivity calculation for the remainder. This pragmatic approach, which essentially combines adjoint methods with finite differences, is presented in the following section. Parts of this theory and the results in this chapter have been published in Refs. [13, 15].

5.2 A problem-adapted efficient computation of the objective gradient

In this section an in-parts adjoint approach is presented that exploits the fact that the computational costs related to the numerical solution of the FBE problem $c_{\mathrm{eq}}(\boldsymbol{\varphi}, q_{\mathrm{eq}}) = 0$ and coordinate transformation $\boldsymbol{q}_{\mathrm{gg}} = \boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}})$ are at least

used here to argument the complexity of the "full adjoint" option, the correct definition and formulation of these derivatives is not considered here.

three orders of magnitude smaller than the cost associated with the plasma edge transport computation.

The derivation starts by isolating the equations that will be treated with the adjoint method from those that will be based on forward sensitivity calculations. q_{eq} and $\boldsymbol{q}_{\mathrm{gg}}$ are thus formally eliminated by introducing the partially reduced objective and model constraints

$$\widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) := \mathcal{J}(\boldsymbol{\varphi}, q_{\mathrm{eq}}(\boldsymbol{\varphi}), \boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi})), \boldsymbol{q}_{\mathrm{pe}}) \tag{5.3}$$

and

$$\widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) := \boldsymbol{c}_{\mathrm{pe}}(\boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi})), \boldsymbol{q}_{\mathrm{pe}}), \tag{5.4}$$

where $q_{\mathrm{eq}}(\boldsymbol{\varphi})$ solves $c_{\mathrm{eq}}(\boldsymbol{\varphi}, q_{\mathrm{eq}}(\boldsymbol{\varphi})) = 0$. With this notation, the constrained optimization problem (4.1) is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\varphi}\in\Phi_{\mathrm{ad}}, \boldsymbol{q}_{\mathrm{pe}}} \quad & \widetilde{\mathcal{J}}\left(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}\right) \\ s.t. \quad & \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) = 0. \end{aligned} \tag{5.5}$$

The partially reduced Lagrangian

$$\widetilde{\mathcal{L}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}}) = \widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) + \left\langle \boldsymbol{q}^*_{\mathrm{pe}}, \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) \right\rangle, \tag{5.6}$$

can then be introduced. Following the derivation of the adjoint approach using the Lagrangian (see section 2.1.2), the directional derivative of the reduced objective function in a direction $\boldsymbol{\delta\varphi}$ is given by

$$\delta\hat{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi}) = \left\langle \partial_{\boldsymbol{\varphi}} \widetilde{\mathcal{L}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}}), \boldsymbol{\delta\varphi} \right\rangle \tag{5.7}$$

$$= \left\langle \partial_{\boldsymbol{\varphi}} \widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})), \boldsymbol{\delta\varphi} \right\rangle + \left\langle \boldsymbol{q}^*_{\mathrm{pe}}, \partial_{\boldsymbol{\varphi}} \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})\, \boldsymbol{\delta\varphi} \right\rangle, \tag{5.8}$$

where now the adjoint variables $\boldsymbol{q}^*_{\mathrm{pe}}(\boldsymbol{\varphi})$ solve the adjoint equation

$$\nabla_{\boldsymbol{q}_{\mathrm{pe}}} \widetilde{\mathcal{L}} = \nabla_{\boldsymbol{q}_{\mathrm{pe}}} \widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) + \partial_{\boldsymbol{q}_{\mathrm{pe}}} \widetilde{\boldsymbol{c}}^*(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})\, \boldsymbol{q}^*_{\mathrm{pe}} = 0 \tag{5.9}$$

and $\boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{\varphi})$ solves $\nabla_{\boldsymbol{q}^*_{\mathrm{pe}}} \widetilde{\mathcal{L}} = \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) = 0$. Since $\partial_{\boldsymbol{q}_{\mathrm{pe}}} \widetilde{\boldsymbol{c}} = \partial_{\boldsymbol{q}_{\mathrm{pe}}} \boldsymbol{c}_{\mathrm{pe}}$ and $\nabla_{\boldsymbol{q}_{\mathrm{pe}}} \widetilde{\mathcal{J}} = \nabla_{\boldsymbol{q}_{\mathrm{pe}}} \mathcal{J}$, one observes that (5.9) are again the adjoint plasma edge transport equations from equation (5.1). However, the derivatives $\partial_{\boldsymbol{\varphi}} \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})$ and $\partial_{\boldsymbol{\varphi}} \widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})$ in (5.8) are not explicitly available. Therefore, the adjoint operator $\partial_{\boldsymbol{\varphi}} \widetilde{\boldsymbol{c}}^*(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})$

cannot be derived and, as a consequence, equation (5.8) cannot be written as an explicit expression for $\nabla\hat{\mathcal{J}}$.

Nevertheless, the entire gradient can be found by using finite difference calculations with $n_{\boldsymbol{\varphi}}$ different perturbation vectors $\boldsymbol{\delta\varphi}$, where each evaluation retrieves a directional derivative

$$\delta\hat{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{\delta\varphi}) \approx \Delta^{\varepsilon}_{\boldsymbol{\varphi}} \widetilde{\mathcal{L}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^{*}_{\mathrm{pe}}), \tag{5.10}$$

with $\boldsymbol{q}_{\mathrm{pe}} = \boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{\varphi})$ and $\boldsymbol{q}^{*}_{\mathrm{pe}} = \boldsymbol{q}^{*}_{\mathrm{pe}}(\boldsymbol{\varphi})$. Here, $\Delta^{\varepsilon}_{\boldsymbol{\varphi}}$ denotes a 'directional' finite difference evaluation with respect to the argument $\boldsymbol{\varphi}$ in direction $\boldsymbol{\delta\varphi}$ and with step size ε. E.g., using a forward finite difference approximation this directional derivative can be evaluated as

$$\begin{aligned}\Delta^{\varepsilon}_{\boldsymbol{\varphi}} \widetilde{\mathcal{L}} &:= \frac{\widetilde{\mathcal{L}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^{*}_{\mathrm{pe}}) - \widetilde{\mathcal{L}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^{*}_{\mathrm{pe}})}{\varepsilon} \\ &= \frac{\widetilde{\mathcal{J}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}, \boldsymbol{q}_{\mathrm{pe}}) - \widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})}{\varepsilon} + \left\langle \boldsymbol{q}^{*}_{\mathrm{pe}}, \frac{\widetilde{\boldsymbol{c}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}, \boldsymbol{q}_{\mathrm{pe}}) - \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})}{\varepsilon} \right\rangle\end{aligned} \tag{5.11}$$

with ε fixed.

In conclusion, the idea is to first successively solve the forward equations to obtain all state variables $q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}$,$\boldsymbol{q}_{\mathrm{pe}}$ and the objective functional $\mathcal{J}$. Afterwards, the adjoint plasma edge transport equations (5.9) are solved for the adjoint plasma variables $\boldsymbol{q}^{*}_{\mathrm{pe}}$. Finally, to get each component $\delta\hat{\mathcal{J}} = \Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{\mathcal{L}}$ of the gradient of the reduced objective function, the perturbed nonlinear problems $c_{\mathrm{eq}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}, q_{\mathrm{eq}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}))$ and $\boldsymbol{q}_{\mathrm{gg}} = \boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}))$ are solved for $\boldsymbol{q}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}))$. The latter can then be used to evaluate the objective function $\widetilde{\mathcal{J}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}, \boldsymbol{q}_{\mathrm{pe}})$ and the plasma residuals $\widetilde{\boldsymbol{c}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}, \boldsymbol{q}_{\mathrm{pe}})$ necessary in expression (5.11). Whereas a straightforward forward difference approximation of $\nabla\hat{\mathcal{J}}$ with $\Delta^{\varepsilon}\hat{\mathcal{J}}$ would lead to $2 \cdot n_{\boldsymbol{\varphi}} + 1$ full simulations, the in-parts adjoint method reduces this to $2 \cdot n_{\boldsymbol{\varphi}} + 1$ magnetic flux and grid evaluations, complemented with one forward and one adjoint plasma edge simulation. Roughly, given the fast evaluations of magnetic flux and gridding, this means the gradient evaluation cost is reduced to twice the computational cost of a forward simulation.

Finally, it should be mentioned that if the forward magnetic and grid generation models would allow for an easy application of AD, one could use a similar

approach to exploit the existing continuous adjoint model. Again, one would first evaluate the forward equations and the adjoint plasma edge transport equations. Finally, the remaining derivative $\nabla_{\boldsymbol{\varphi}}\widetilde{\mathcal{L}}$ could be evaluated through the use of AD software on a function evaluating $\widetilde{\mathcal{L}}$ with $\boldsymbol{q}$ and $\boldsymbol{q}^*$ following from forward and adjoint plasma edge simulations.

5.3 Introducing the continuous adjoint plasma edge model

Before interpreting the introduced approach in a discrete model setting, the continuous adjoint plasma edge equations

$$\partial_{\boldsymbol{q}_{\mathrm{pe}}}\widetilde{\boldsymbol{c}}^*(\boldsymbol{\varphi},\boldsymbol{q}_{\mathrm{pe}})\,\boldsymbol{q}_{\mathrm{pe}}^* = -\nabla_{\boldsymbol{q}_{\mathrm{pe}}}\widetilde{\mathcal{J}}(\boldsymbol{\varphi},\boldsymbol{q}_{\mathrm{pe}}) \tag{5.12}$$

are derived in detail. As explained in section 2.1.2, the first step to obtain the adjoint equations (5.12) consists of linearizing the transport equations $\left\langle \boldsymbol{q}_{\mathrm{pe}}^*, \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi},\boldsymbol{q}_{\mathrm{pe}})\right\rangle$, with respect to the plasma state $\boldsymbol{q}_{\mathrm{pe}}$. Then, the linear differential operator can then be 'pulled' to the primal space using the adjoint identity (2.9). Given the complex set of plasma edge equations and boundary conditions introduced in section 3.3, this might sound easier than it is. Especially finding the adjoint boundary conditions might be challenging using an approach as for example in Refs. [44, 58, 145]. Therefore, a procedure called the *formal Lagrangian approach* is followed here to derive the adjoint equations [21, 141].

This approach starts from the partially reduced Lagrangian in equation (5.6). Many choices of the inner product $\langle\cdot,\cdot\rangle$ in the Lagrangian definition are possible. Since the appropriate inner product is not known in general applications, the formal Lagrangian approach assumes standard L^2-type inner products. Recall the definitions of the domain contribution of the plasma edge equations $\boldsymbol{\mathcal{B}}(\boldsymbol{q}_{\mathrm{gg}},\boldsymbol{q}_{\mathrm{pe}}) = 0$ and its boundary conditions $\boldsymbol{\mathcal{C}}(\boldsymbol{q}_{\mathrm{gg}},\boldsymbol{q}_{\mathrm{pe}}) = 0$ that jointly form the plasma edge equations $\boldsymbol{c}_{\mathrm{pe}} = (\boldsymbol{\mathcal{B}}^\top, \boldsymbol{\mathcal{C}}^\top)^\top = 0$. Using separate multipliers $\boldsymbol{p}^*$ for the boundary conditions $\boldsymbol{\mathcal{C}}$ on $\Sigma = \partial\Omega_{\mathrm{pe}}$, the inner product for the plasma edge equations is thus chosen as

$$\left\langle \boldsymbol{q}_{\mathrm{pe}}^*, \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi},\boldsymbol{q}_{\mathrm{pe}})\right\rangle := \left\langle \boldsymbol{q}_{\mathrm{pe}}^*, \widetilde{\boldsymbol{\mathcal{B}}}(\boldsymbol{\varphi},\boldsymbol{q}_{\mathrm{pe}})\right\rangle_{\Omega_{\mathrm{pe}}} + \left\langle \boldsymbol{p}^*, \widetilde{\boldsymbol{\mathcal{C}}}(\boldsymbol{\varphi},\boldsymbol{q}_{\mathrm{pe}})\right\rangle_{\Sigma}. \tag{5.13}$$

Here, the domain and boundary inner products are defined as

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\Omega_{\mathrm{pe}}} := \int_{\Omega_{\mathrm{pe}}} \boldsymbol{a} \cdot \boldsymbol{b} \, \mathrm{d}\boldsymbol{\omega}, \tag{5.14}$$

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\Sigma} := \int_{\Sigma} \boldsymbol{a} \cdot \boldsymbol{b} \, \mathrm{d}\sigma, \tag{5.15}$$

where $\widetilde{\mathcal{B}}$ and $\widetilde{\mathfrak{C}}$ are partially reduced forms of $\mathcal{B}$ and $\mathfrak{C}$ that eliminate $\boldsymbol{q}_{\mathrm{gg}}$ and q_{eq}. Like the state variables, each state equation has its own adjoint variable $\boldsymbol{q}^*_{\mathrm{pe}} = (n^*, u^*_{\parallel}, p^*_{\mathrm{n}}, T^*)^T$. Likewise, each boundary condition obtains an adjoint variable, $\boldsymbol{p}^* = (n^*_S, u^*_{\parallel,S}, p^*_{\mathrm{n},S}, T^*_S)^T$. Then, notice that the cost functional presented earlier can be represented in the generic form

$$\widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) = \int_{\Sigma} \widetilde{\mathcal{F}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) \, d\sigma + g(\boldsymbol{\varphi}), \tag{5.16}$$

where the integrand $\widetilde{\mathcal{F}}$ and the term g are again appropriately reduced to only depend on $\boldsymbol{\varphi}$ and $\boldsymbol{q}$. Notice that only objective functions on the boundary are considered here. The extension to include domain terms is straightforward. For different types of objective functionals and the corresponding changes in adjoint plasma edge equations, the reader is referred to [41]. Combining these components, the Lagrangian is finally given as

$$\begin{aligned} \widetilde{\mathcal{L}} &= \widetilde{\mathcal{J}} + \left\langle \boldsymbol{q}^*_{\mathrm{pe}}, \widetilde{\boldsymbol{c}} \right\rangle \\ &= \int_{\Omega_{\mathrm{pe}}} \boldsymbol{q}^*_{\mathrm{pe}} \cdot \widetilde{\mathcal{B}} \, \mathrm{d}\boldsymbol{\omega} + \int_{\Sigma} \left(\widetilde{\mathcal{F}} + \boldsymbol{p}^* \cdot \widetilde{\mathfrak{C}} \right) \mathrm{d}\sigma + g, \end{aligned}$$

5.3.1 Linearization

Following the derivation in Dekeyser [41], the formal Lagrangian approach will now be applied to the flexible convection-diffusion form of the plasma edge equations for generality. First, the Lagrangian is linearized with respect to $\boldsymbol{q}_{\mathrm{pe}}$. Using subscript $(\cdot)_{\boldsymbol{q}}$ as a shorthand notation for the differentiation with respect to $\boldsymbol{q}_{\mathrm{pe}}$, this linearization is given by

$$\widetilde{\mathcal{L}}_{\boldsymbol{q}} \delta \boldsymbol{q}_{\mathrm{pe}} = \int_{\Omega_{\mathrm{pe}}} \boldsymbol{q}^*_{\mathrm{pe}} \cdot \widetilde{\mathcal{B}}_{\boldsymbol{q}} \, \delta \boldsymbol{q}_{\mathrm{pe}} \, \mathrm{d}\boldsymbol{\omega} + \int_{\Sigma} \left(\widetilde{\mathcal{F}}_{\boldsymbol{q}} + \boldsymbol{p}^* \cdot \widetilde{\mathfrak{C}}_{\boldsymbol{q}} \right) \delta \boldsymbol{q}_{\mathrm{pe}} \, \mathrm{d}\sigma. \tag{5.17}$$

Now, the components of this equation can be identified one by one.

The linearized form of the state equations is given by

$$\begin{aligned}\widetilde{\mathcal{B}}_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} = {} & S_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} + S_{\nabla_\theta \boldsymbol{q}}\frac{1}{h_\theta}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial\theta} + S_{\nabla_r \boldsymbol{q}}\frac{1}{h_r}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial r} \\ & - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}C^\theta_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} - \frac{\sqrt{g}}{h_\theta^2}\left(D^\theta\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial\theta} + D^\theta_{\boldsymbol{q}}\frac{\partial\boldsymbol{q}_{\mathrm{pe}}}{\partial\theta}\delta\boldsymbol{q}_{\mathrm{pe}}\right)\right) \\ & - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}C^r_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} - \frac{\sqrt{g}}{h_r^2}\left(D^r\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial r} + D^r_{\boldsymbol{q}}\frac{\partial\boldsymbol{q}_{\mathrm{pe}}}{\partial r}\delta\boldsymbol{q}_{\mathrm{pe}}\right)\right),\end{aligned}$$

where subscripts $\nabla_\theta\boldsymbol{q}$ and $\nabla_r\boldsymbol{q}$ denote a differentiation with respect to the occurrence of poloidal or radial gradients of $\boldsymbol{q}_{\mathrm{pe}}$, respectively[†]. Similarly, the occurrence of poloidal and radial gradients in the objective functional and boundary conditions can be accounted for by representing them with the functions $f(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \nabla_\theta\boldsymbol{q}_{\mathrm{pe}}, \nabla_r\boldsymbol{q}_{\mathrm{pe}}) = 0$ and $\mathcal{C}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \nabla_\theta\boldsymbol{q}_{\mathrm{pe}}, \nabla_r\boldsymbol{q}_{\mathrm{pe}}) = 0$, with

$$\begin{aligned}\widetilde{\mathcal{C}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) &:= \mathcal{C}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \nabla_\theta\boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{pe}}), \nabla_r\boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{pe}})) = 0, \\ \widetilde{\mathcal{F}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) &:= f(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \nabla_\theta\boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{pe}}), \nabla_r\boldsymbol{q}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{pe}})) = 0.\end{aligned}$$

The linearized objective and boundary conditions can then be written as

$$\begin{aligned}\widetilde{\mathcal{F}}_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} &= f_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} + f_{\nabla_\theta\boldsymbol{q}}\frac{1}{h_\theta}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial\theta} + f_{\nabla_r\boldsymbol{q}}\frac{1}{h_r}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial r} = 0, \\ \widetilde{\mathcal{C}}_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} &= \mathcal{C}_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} + \mathcal{C}_{\nabla_\theta\boldsymbol{q}}\frac{1}{h_\theta}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial\theta} + \mathcal{C}_{\nabla_r\boldsymbol{q}}\frac{1}{h_r}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial r} = 0.\end{aligned}$$

5.3.2 Integration by parts

The next step of the formal Lagrangian approach then consists in applying integration by parts on the domain integral. Hence, one obtains

$$\begin{aligned}\widetilde{\mathcal{L}}_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} &= \int_{\Omega_{\mathrm{pe}}}\left(\widetilde{\mathcal{B}}^*_{\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}}\right)\cdot\delta\boldsymbol{q}_{\mathrm{pe}}\,\mathrm{d}\boldsymbol{\omega} + \int_\Sigma\left(\widetilde{\mathcal{F}}_{\boldsymbol{q}} + \boldsymbol{p}^*\cdot\widetilde{\mathcal{C}}_{\boldsymbol{q}} + \mathrm{BT}\right)\delta\boldsymbol{q}_{\mathrm{pe}}\,\mathrm{d}\sigma, \\ &= \left\langle\widetilde{\mathcal{B}}^*_{\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}}, \delta\boldsymbol{q}_{\mathrm{pe}}\right\rangle_{\Omega_{\mathrm{pe}}} + \int_\Sigma\left(\widetilde{\mathcal{F}}_{\boldsymbol{q}} + \boldsymbol{p}^*\cdot\widetilde{\mathcal{C}}_{\boldsymbol{q}} + \mathrm{BT}\right)\delta\boldsymbol{q}_{\mathrm{pe}}\,\mathrm{d}\sigma,\end{aligned}$$

[†]The linearized diffusion coefficients are defined here as $(D^\theta_{\boldsymbol{q}})_{i,j,k} := \partial_{q_j}D^\theta_{i,k}$ and $(D^r_{\boldsymbol{q}})_{i,j,k} := \partial_{q_j}D^r_{i,k}$ for convenience of the subsequent derivation.

with $\widetilde{\mathcal{B}}^*_{\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}}$ given by [41]

$$\begin{aligned}\widetilde{\mathcal{B}}^*_{\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}} = {} & S^\top_{\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}} - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}S^\top_{\nabla_\theta\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}}\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}S^\top_{\nabla_r\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}}\right) \\ & + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}(C^\theta_{\boldsymbol{q}})^\top\boldsymbol{q}^*_{\mathrm{pe}} + \frac{\sqrt{g}}{h_\theta^2}(D^\theta)^\top\frac{\partial\boldsymbol{q}^*_{\mathrm{pe}}}{\partial\theta}\right) \\ & + \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}(C^r_{\boldsymbol{q}})^\top\boldsymbol{q}^*_{\mathrm{pe}} + \frac{\sqrt{g}}{h_r^2}(D^r)^\top\frac{\partial\boldsymbol{q}^*_{\mathrm{pe}}}{\partial r}\right) \\ & - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\theta}\left(\frac{\sqrt{g}}{h_\theta}(C^\theta_{\boldsymbol{q}})^\top\right)\boldsymbol{q}^*_{\mathrm{pe}} - \frac{1}{\sqrt{g}}\frac{\partial}{\partial r}\left(\frac{\sqrt{g}}{h_r}(C^r_{\boldsymbol{q}})^\top\right)\boldsymbol{q}^*_{\mathrm{pe}} \\ & - \frac{1}{h_\theta}\frac{\partial(\boldsymbol{q}_{\mathrm{pe}})^\top}{\partial\theta}(D^\theta_{\boldsymbol{q}})^\top\frac{1}{h_\theta}\frac{\partial\boldsymbol{q}^*_{\mathrm{pe}}}{\partial\theta} - \frac{1}{h_r}\frac{\partial(\boldsymbol{q}_{\mathrm{pe}})^\top}{\partial r}(D^r_{\boldsymbol{q}})^\top\frac{1}{h_r}\frac{\partial\boldsymbol{q}^*_{\mathrm{pe}}}{\partial r}.\end{aligned} \tag{5.18}$$

and

$$\begin{aligned}\mathrm{BT}\,\delta\boldsymbol{q}_{\mathrm{pe}} = {} & \boldsymbol{q}^*_{\mathrm{pe}}\cdot(S_{\nabla_\theta\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}})\nu_\theta + \boldsymbol{q}^*_{\mathrm{pe}}\cdot(S_{\nabla_r\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}})\nu_r \\ & - \boldsymbol{q}^*_{\mathrm{pe}}\cdot\left(C^\theta_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} - D^\theta\frac{1}{h_\theta}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial\theta} - D^\theta_{\boldsymbol{q}}\frac{1}{h_\theta}\frac{\partial\boldsymbol{q}_{\mathrm{pe}}}{\partial\theta}\delta\boldsymbol{q}_{\mathrm{pe}}\right)\nu_\theta \\ & - \boldsymbol{q}^*_{\mathrm{pe}}\cdot\left(C^r_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}} - D^r\frac{1}{h_r}\frac{\partial\delta\boldsymbol{q}_{\mathrm{pe}}}{\partial r} - D^r_{\boldsymbol{q}}\frac{1}{h_r}\frac{\partial\boldsymbol{q}_{\mathrm{pe}}}{\partial r}\delta\boldsymbol{q}_{\mathrm{pe}}\right)\nu_r \\ & - \frac{1}{h_\theta}\frac{\partial\boldsymbol{q}^*_{\mathrm{pe}}}{\partial\theta}\cdot\left(D^\theta\delta\boldsymbol{q}_{\mathrm{pe}}\right)\nu_\theta - \frac{1}{h_r}\frac{\partial\boldsymbol{q}^*_{\mathrm{pe}}}{\partial r}\cdot\left(D^r\delta\boldsymbol{q}_{\mathrm{pe}}\right)\nu_r\end{aligned} \tag{5.19}$$

the boundary terms resulting from the integration.

Equating $\widetilde{\mathcal{L}}_{\boldsymbol{q}}\delta\boldsymbol{q}_{\mathrm{pe}}$ to zero for all possible functions $\delta\boldsymbol{q}_{\mathrm{pe}}$ then leads to the adjoint equations. The adjoint plasma flow equations are thus given by the domain contribution $\widetilde{\mathcal{B}}^*_{\boldsymbol{q}}\boldsymbol{q}^*_{\mathrm{pe}} = 0$. The strong resemblance of these adjoint equations on the forward equations (3.30) allows reusing the numeric routines of the forward code to solve the adjoint equations. In comparison to the forward convection-diffusion equations, additional sources arise due to the state-dependent transport coefficients. More important, it should be noted that convection terms have switched signs. This supports interpreting the adjoint flow equations as a set of equations that propagate objective information in the reverse direction. Indeed, it will be seen further that the linearized objective term serves as a source in the adjoint boundary conditions.

5.3.3 Adjoint boundary conditions

The adjoint boundary conditions are found by additionally equating the integrand of the surface integral to zero for all possible perturbations $\delta \boldsymbol{q}_{\mathrm{pe}}$. Isolating the perturbations $\delta \boldsymbol{q}_{\mathrm{pe}}$, $\frac{1}{h_\theta}\frac{\partial}{\partial \theta}\delta \boldsymbol{q}_{\mathrm{pe}}$ and $\frac{1}{h_r}\frac{\partial}{\partial r}\delta \boldsymbol{q}_{\mathrm{pe}}$ in the integrand gives

$$
\begin{aligned}
& f_{\boldsymbol{q}}\delta \boldsymbol{q}_{\mathrm{pe}} + \boldsymbol{p}^* \cdot \left(\mathcal{C}_{\boldsymbol{q}}\delta \boldsymbol{q}_{\mathrm{pe}} + \mathcal{C}_{\nabla_\theta \boldsymbol{q}} \frac{1}{h_\theta}\frac{\partial \delta \boldsymbol{q}_{\mathrm{pe}}}{\partial \theta} + \mathcal{C}_{\nabla_r \boldsymbol{q}} \frac{1}{h_r}\frac{\partial \delta \boldsymbol{q}_{\mathrm{pe}}}{\partial r} \right) + \mathrm{BT}\, \delta \boldsymbol{q}_{\mathrm{pe}} \\
& = \left(\nabla_{\boldsymbol{q}} f + \mathcal{C}_{\boldsymbol{q}}^\top \boldsymbol{p}^* + \nu_\theta S_{\nabla_\theta \boldsymbol{q}}^\top \boldsymbol{q}_{\mathrm{pe}}^* + \nu_r S_{\nabla_r \boldsymbol{q}}^\top \boldsymbol{q}_{\mathrm{pe}}^* \right) \cdot \delta \boldsymbol{q}_{\mathrm{pe}} \\
& - \nu_\theta \left((C_{\boldsymbol{q}}^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^* - \frac{1}{h_\theta}\frac{\partial (\boldsymbol{q}_{\mathrm{pe}})^\top}{\partial \theta}(D_{\boldsymbol{q}}^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^* + (D^\theta)^\top \frac{1}{h_\theta}\frac{\partial \boldsymbol{q}_{\mathrm{pe}}^*}{\partial \theta} \right) \cdot \delta \boldsymbol{q}_{\mathrm{pe}} \\
& - \nu_r \left((C_{\boldsymbol{q}}^r)^\top \boldsymbol{q}_{\mathrm{pe}}^* - \frac{1}{h_r}\frac{\partial (\boldsymbol{q}_{\mathrm{pe}})^\top}{\partial r}(D_{\boldsymbol{q}}^r)^\top \boldsymbol{q}_{\mathrm{pe}}^* + (D^r)^\top \frac{1}{h_r}\frac{\partial \boldsymbol{q}_{\mathrm{pe}}^*}{\partial r} \right) \cdot \delta \boldsymbol{q}_{\mathrm{pe}} \\
& + \left(f_{\nabla_\theta \boldsymbol{q}}^\top + \mathcal{C}_{\nabla_\theta \boldsymbol{q}}^\top \boldsymbol{p}^* + \nu_\theta (D^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^* \right) \cdot \frac{1}{h_\theta}\frac{\partial \delta \boldsymbol{q}_{\mathrm{pe}}}{\partial \theta} \\
& + \left(f_{\nabla_r \boldsymbol{q}}^\top + \mathcal{C}_{\nabla_r \boldsymbol{q}}^\top \boldsymbol{p}^* + \nu_r (D^r)^\top \boldsymbol{q}_{\mathrm{pe}}^* \right) \cdot \frac{1}{h_r}\frac{\partial \delta \boldsymbol{q}_{\mathrm{pe}}}{\partial r}
\end{aligned}
\tag{5.20}
$$

Further manipulation of this equation by integrating the last two terms by parts allows isolating $\delta \boldsymbol{q}_{\mathrm{pe}}$ and obtain the adjoint equations

$$
\begin{aligned}
0 = \widetilde{\mathcal{L}}_{\boldsymbol{q}} \delta \boldsymbol{q}_{\mathrm{pe}} &= \left\langle \widetilde{\mathcal{B}}_{\boldsymbol{q}}^* \boldsymbol{q}_{\mathrm{pe}}^*, \delta \boldsymbol{q}_{\mathrm{pe}} \right\rangle_{\Omega_{\mathrm{pe}}} + \int_\Sigma \left(\widetilde{\mathcal{F}}_{\boldsymbol{q}} + \boldsymbol{p}^* \cdot \widetilde{\mathcal{C}}_{\boldsymbol{q}} + \mathrm{BT} \right) \delta \boldsymbol{q}_{\mathrm{pe}}\, \mathrm{d}\sigma, \\
&= \left\langle \widetilde{\mathcal{B}}_{\boldsymbol{q}}^* \boldsymbol{q}_{\mathrm{pe}}^*, \delta \boldsymbol{q}_{\mathrm{pe}} \right\rangle_{\Omega_{\mathrm{pe}}} + \left\langle \nabla_{\boldsymbol{q}} \widetilde{\mathcal{F}} + \widetilde{\mathcal{C}}_{\boldsymbol{q}}^* \boldsymbol{p}^* + \mathrm{BT}^*, \delta \boldsymbol{q}_{\mathrm{pe}} \right\rangle_\Sigma ,
\end{aligned}
$$

However, the adjoint boundary conditions can equally be found through imposing in equation (5.20) directly that the multiplication factors of $\delta \boldsymbol{q}_{\mathrm{pe}}$, $\frac{1}{h_\theta}\frac{\partial}{\partial \theta}\delta \boldsymbol{q}_{\mathrm{pe}}$ and $\frac{1}{h_r}\frac{\partial}{\partial r}\delta \boldsymbol{q}_{\mathrm{pe}}$ equal zero simultaneously and the integrand of the surface integral thus also equals zero:

$$
\begin{cases}
0 = \nabla_{\boldsymbol{q}} f + \mathcal{C}_{\boldsymbol{q}}^\top \boldsymbol{p}^* + \nu_\theta S_{\nabla_\theta \boldsymbol{q}}^\top \boldsymbol{q}_{\mathrm{pe}}^* + \nu_r S_{\nabla_r \boldsymbol{q}}^\top \boldsymbol{q}_{\mathrm{pe}}^* \\
\quad - \nu_\theta \left((C_{\boldsymbol{q}}^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^* - \frac{1}{h_\theta}\frac{\partial (\boldsymbol{q}_{\mathrm{pe}})^\top}{\partial \theta}(D_{\boldsymbol{q}}^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^* + (D^\theta)^\top \frac{1}{h_\theta}\frac{\partial \boldsymbol{q}_{\mathrm{pe}}^*}{\partial \theta} \right) \\
\quad - \nu_r \left((C_{\boldsymbol{q}}^r)^\top \boldsymbol{q}_{\mathrm{pe}}^* - \frac{1}{h_r}\frac{\partial (\boldsymbol{q}_{\mathrm{pe}})^\top}{\partial r}(D_{\boldsymbol{q}}^r)^\top \boldsymbol{q}_{\mathrm{pe}}^* + (D^r)^\top \frac{1}{h_r}\frac{\partial \boldsymbol{q}_{\mathrm{pe}}^*}{\partial r} \right), \\
0 = f_{\nabla_\theta \boldsymbol{q}}^\top + \mathcal{C}_{\nabla_\theta \boldsymbol{q}}^\top \boldsymbol{p}^* + \nu_\theta (D^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^*, \\
0 = f_{\nabla_r \boldsymbol{q}}^\top + \mathcal{C}_{\nabla_r \boldsymbol{q}}^\top \boldsymbol{p}^* + \nu_r (D^r)^\top \boldsymbol{q}_{\mathrm{pe}}^*.
\end{cases}
\tag{5.21}
$$

As discussed in [41], one of the last two sets of equations is redundant. The two remaining sets of equations give the adjoint boundary conditions and a relation between the boundary multipliers $\boldsymbol{p}^*$ and the domain multipliers at the edge $\boldsymbol{q}_{\mathrm{pe}}$. To reuse the forward code again, these adjoint boundary conditions are reformulated finally into the form

$$\boldsymbol{\Gamma}_{\mathrm{b}}^* \cdot \boldsymbol{\nu} = S_{\mathrm{c}} + S_{\mathrm{v}}\phi,$$

in analogy to the forward boundary equation implementation (see section 3.3.4), with $\boldsymbol{\Gamma}_{\mathrm{b}}^*$ the adjoint flux of the considered equation at the boundary.

5.4 Derivation of a semi-discrete approach to in-parts adjoint sensitivity calculation

Remark that in the introduction of the in-parts adjoint approach, discretization effects were not considered. However, the finite differences suggest differentiation of the discrete code, whereas the available adjoint plasma code is a continuous adjoint code. This section will elaborate some considerations on the influence of discretization effects in order to develop a practical implementation.

The boundary conditions in the (still continuous) adjoint equation and sensitivity of section 5.2 are first isolated using the inner product definitions (5.14) and (5.15). The adjoint equations are then given along with their boundary conditions as

$$0 = \partial_{\boldsymbol{q}_{\mathrm{pe}}}\widetilde{\mathcal{J}}\,\delta\boldsymbol{q}_{\mathrm{pe}} + \left\langle \widetilde{\boldsymbol{\mathcal{B}}}_{\boldsymbol{q}}^{*}\boldsymbol{q}_{\mathrm{pe}}^{*}, \delta\boldsymbol{q}_{\mathrm{pe}} \right\rangle_{\Omega_{\mathrm{pe}}} + \left\langle \widetilde{\boldsymbol{\mathcal{C}}}_{\boldsymbol{q}}^{*}\boldsymbol{p}^{*} + \mathrm{BT}^{*}, \delta\boldsymbol{q}_{\mathrm{pe}} \right\rangle_{\Sigma}$$

and the sensitivity by

$$\delta\hat{\mathcal{J}} = \partial_{\boldsymbol{\varphi}}\widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})\boldsymbol{\delta\varphi} + \left\langle \boldsymbol{q}_{\mathrm{pe}}^{*}, \widetilde{\boldsymbol{\mathcal{B}}}_{\boldsymbol{\varphi}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})\boldsymbol{\delta\varphi} \right\rangle_{\Omega_{\mathrm{pe}}} + \left\langle \boldsymbol{p}^{*}, \widetilde{\boldsymbol{\mathcal{C}}}_{\boldsymbol{\varphi}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})\boldsymbol{\delta\varphi} \right\rangle_{\Sigma}.$$

In this equation, the subscript $(\cdot)_{\boldsymbol{\varphi}}$ indicates a differentiation with respect to the design variables. Using the discretization operator $\{\cdot\}_{\mathcal{H}}^{i}$ to denote the i^{th} element of a discretization with characteristic distance $\mathcal{H}$, the discretized continuous adjoint equations read

$$0 = \sum_i \left\{ \widetilde{\boldsymbol{\mathcal{B}}}_{\boldsymbol{q}}^{*} \right\}_{\mathcal{H}}^{i} \boldsymbol{q}_{\mathrm{pe},\mathcal{H},\boldsymbol{i}}^{*}\Omega_{\mathrm{pe}}^{i} + \sum_j \left(\left\{ \widetilde{\mathcal{F}}_{\boldsymbol{q}} \right\}_{\mathcal{H}}^{j} + \left\{ \widetilde{\boldsymbol{\mathcal{C}}}_{\boldsymbol{q}}^{*} \right\}_{\mathcal{H}}^{j} \boldsymbol{p}_{\mathcal{H},\boldsymbol{j}}^{*} + \{\mathrm{BT}^{*}\}_{\mathcal{H}}^{j} \right) \Sigma^{j}, \quad (5.22)$$

where the spatial integration over domain and boundary in the definition of the continuous inner products is replaced by discrete summations over the finite volumes Ω^i_{pe} and boundary elements Σ^j. Similarly, discretizing the sensitivity gives

$$\delta\hat{\mathcal{J}}_{\mathcal{H}} = \sum_i (\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{i}})^\top \left\{\widetilde{\boldsymbol{\mathcal{B}}}_{\boldsymbol{\varphi}}\right\}^i_{\mathcal{H}} \boldsymbol{\delta\varphi}\Omega^i_{\mathrm{pe}} + \sum_j \left(\left\{\widetilde{\mathcal{F}}_{\boldsymbol{\varphi}}\right\}^j_{\mathcal{H}} + (\boldsymbol{p}^*_{\mathcal{H},\boldsymbol{i}})^\top \left\{\widetilde{\boldsymbol{\mathcal{C}}}_{\boldsymbol{\varphi}}\right\}^j_{\mathcal{H}}\right) \boldsymbol{\delta\varphi}\Sigma^j. \tag{5.23}$$

Remark that since a discrete control vector $\boldsymbol{\varphi}$ was chosen in the last chapter, the control vector is not discretized. Otherwise, the control function needs to be discretized as well. The first step to find the sensitivity $\delta\hat{\mathcal{J}}_{\mathcal{H}}$ is thus to numerically solve the discretized adjoint equations (5.22) for $\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{i}}$ and $\boldsymbol{p}^*_{\mathcal{H},\boldsymbol{i}}$ and plugging these values into the sensitivity expression (5.23).

Towards a semi-discrete approach to sensitivity calculation

The question of interest for the in-parts adjoint approach is if the discretized directional derivatives $\left\{\widetilde{\mathcal{F}}_{\boldsymbol{\varphi}}\right\}_{\mathcal{H}} \boldsymbol{\delta\varphi}$ and more importantly $\left\{\widetilde{\boldsymbol{\mathcal{B}}}_{\boldsymbol{\varphi}}\right\}_{\mathcal{H}} \boldsymbol{\delta\varphi}$ and $\left\{\widetilde{\boldsymbol{\mathcal{C}}}_{\boldsymbol{\varphi}}\right\}_{\mathcal{H}} \boldsymbol{\delta\varphi}$ can be approximated by finite differencing the discrete code. Note that using finite difference evaluations on the actual code will resort in an approximation of the discrete derivatives $\partial_{\boldsymbol{\varphi}}\widetilde{\mathcal{F}}_h\boldsymbol{\delta\varphi}$, $\partial_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{B}}}_h\boldsymbol{\delta\varphi}$, and $\partial_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{C}}}_h\boldsymbol{\delta\varphi}$, with $\widetilde{\mathcal{F}}_h$ the discrete objective function, and $\widetilde{\boldsymbol{\mathcal{B}}}_h$ and $\widetilde{\boldsymbol{\mathcal{C}}}_h$ the discretized equations and boundary conditions. Since these are components of the discrete optimization problem, a mix between the discretize-then-optimize (DTO) and optimize-then-discretize (OTD) approach is thus targeted (see discussion on discrete and continuous adjoints in section 2.1.2). The question therefore arises when these DTO sensitivities can substitute their OTD equivalents in the sensitivity expression (5.23). Of course, an evident requirement is that the discretization should be executed on the same grid. Obviously, they can only match up till the level of discretization errors. Given these two remarks, the approximation of the objective function term $\left\{\widetilde{\mathcal{F}}_{\boldsymbol{\varphi}}\right\}_{\mathcal{H}} \boldsymbol{\delta\varphi} \approx \partial_{\boldsymbol{\varphi}}\widetilde{\mathcal{F}}_h\boldsymbol{\delta\varphi}$ is a reasonable assumption.

However, for $\left\{\widetilde{\boldsymbol{\mathcal{B}}}_{\boldsymbol{\varphi}}\right\}_{\mathcal{H}} \boldsymbol{\delta\varphi}$ and $\left\{\widetilde{\boldsymbol{\mathcal{C}}}_{\boldsymbol{\varphi}}\right\}_{\mathcal{H}} \boldsymbol{\delta\varphi}$, the situation needs to be assessed with more care. First of all, these sensitivities depend on the inner product choice at the continuous level [21]. The inner products on discrete and continuous level should therefore be chosen consistently. Moreover, it should be noted that any

scaling of $\widetilde{\mathcal{B}} = 0$ and $\widetilde{\mathfrak{C}} = 0$ with a constant multiplier will yield the same state variables. Also the sensitivity will not change, since the adjoint variables from the adjoint equations (5.22) will adjust accordingly. However, it should be made sure that the model equations and boundary conditions in the discrete code $\widetilde{\mathcal{B}}_h = 0$ and $\widetilde{\mathfrak{C}}_h = 0$ are scaled in the same way as the continuous expressions $\widetilde{\mathcal{B}} = 0$ and $\widetilde{\mathfrak{C}} = 0$ used in the continuous adjoint derivation. For the same reason, they should be expressed in the same dimensions.

For the domain contribution, this is naturally fulfilled. These equations are both derived and implemented in the convection-diffusion form. For the boundary conditions, however, this requirement is often not fulfilled. Indeed, looking back into the boundary formulations in chapter 3, it can be seen that these boundary conditions are often expressed in a convenient way. For example $n_\mathrm{c} - n_\mathrm{i} = 0$ represents the Dirichlet boundary condition at the core for the plasma continuity equation. In contrast, the same boundary condition in the discrete code will be imposed in the form

$$\boldsymbol{\Gamma}_\mathrm{b} \cdot \boldsymbol{\nu} = -\mathbb{B}n_\mathrm{c} + \mathbb{B}n_\mathrm{i},$$

with the dimension of a plasma flux. It is clear that if finite differences are used to find the sensitivity of the code's boundary conditions $\widetilde{\mathcal{B}}_h$, the discretized boundary multipliers $\boldsymbol{p}^*_{\mathcal{H}}$ will not be their correct adjoint multipliers. The derivation of the adjoint boundary conditions should therefore be done such that they match the discrete boundary conditions.

Finding the correct boundary multipliers

Consistent boundary multipliers $\boldsymbol{p}^*_{\mathcal{H}}$ for the discrete implementation will now be derived. First, the structure of the discrete code is accounted for by writing the boundary conditions as

$$\begin{aligned}\widetilde{\mathfrak{C}}(\boldsymbol{\varphi}, \boldsymbol{q}_\mathrm{pe}) = \mathcal{S}(\boldsymbol{\varphi}, \boldsymbol{q}_\mathrm{pe}, \nabla_\theta \boldsymbol{q}_\mathrm{pe}, \nabla_r \boldsymbol{q}_\mathrm{pe}) \\ + \nu_\theta \left(C^\theta - D^\theta \frac{1}{h_\theta} \frac{\partial \boldsymbol{q}_\mathrm{pe}}{\partial \theta} \right) + \nu_r \left(C^r - D^r \frac{1}{h_r} \frac{\partial \boldsymbol{q}_\mathrm{pe}}{\partial r} \right),\end{aligned} \tag{5.24}$$

where the vector $\mathcal{S}$ includes the boundary condition sources imposed in the guard cell. Linearizing these boundary conditions to $\boldsymbol{q}_\mathrm{pe}, \nabla_\theta \boldsymbol{q}_\mathrm{pe}, \nabla_r \boldsymbol{q}_\mathrm{pe}$ and substitution

in (5.21) leads to

$$\begin{cases} \nu_\theta \left((C_{\boldsymbol{q}}^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^* + (D^\theta)^\top \frac{1}{h_\theta} \frac{\partial \boldsymbol{q}_{\mathrm{pe}}^*}{\partial \theta} \right) + \nu_r \left((C_{\boldsymbol{q}}^r)^\top \boldsymbol{q}_{\mathrm{pe}}^* + (D^r)^\top \frac{1}{h_r} \frac{\partial \boldsymbol{q}_{\mathrm{pe}}^*}{\partial r} \right) \\ \quad = \nabla_{\boldsymbol{q}} f + \mathcal{S}_{\boldsymbol{q}}^\top \boldsymbol{p}^* + \nu_\theta \left((C_{\boldsymbol{q}}^\theta)^\top \boldsymbol{p}^* + S_{\nabla_\theta \boldsymbol{q}}^\top \boldsymbol{q}_{\mathrm{pe}}^* \right) + \nu_r \left((C_{\boldsymbol{q}}^r)^\top \boldsymbol{p}^* + S_{\nabla_r \boldsymbol{q}}^\top \boldsymbol{q}_{\mathrm{pe}}^* \right), \\ 0 = f_{\nabla_\theta \boldsymbol{q}}^\top + \mathcal{S}_{\nabla_\theta \boldsymbol{q}}^\top \boldsymbol{p}^* - \nu_\theta (D^\theta)^\top \boldsymbol{p}^* + \nu_\theta (D^\theta)^\top \boldsymbol{q}_{\mathrm{pe}}^*, \\ 0 = f_{\nabla_r \boldsymbol{q}}^\top + \mathcal{S}_{\nabla_r \boldsymbol{q}}^\top \boldsymbol{p}^* - \nu_r (D^r)^\top \boldsymbol{p}^* + \nu_r (D^r)^\top \boldsymbol{q}_{\mathrm{pe}}^*. \end{cases} \tag{5.25}$$

Note that, since all terms associated to the boundary conditions $\widetilde{\mathfrak{C}} = 0$ are scaled with a separate multiplier, scaling a boundary condition will only affect the boundary multiplier $\boldsymbol{p}^*$. The adjoint boundary conditions themselves follow from eliminating $\boldsymbol{p}^*$ from the system (5.25) and are therefore not influenced by this change. The adjoint boundary conditions from Dekeyser [41] can therefore largely be reused (except for some new boundary condition terms).

The $\boldsymbol{p}^*$ following from this particular choice for $\widetilde{\mathfrak{C}}$ is now the correct multiplier for the code's boundary conditions. It can be seen that the solution of equations (5.25) for $\boldsymbol{p}^*$ would be significantly simplified if $f_{\nabla_\theta \boldsymbol{q}} = 0$, $f_{\nabla_r \boldsymbol{q}} = 0$, $\mathcal{S}_{\nabla_\theta \boldsymbol{q}} = 0$, and $\mathcal{S}_{\nabla_r \boldsymbol{q}} = 0$. In that case, $\boldsymbol{p}^* = \boldsymbol{q}_{\mathrm{pe}}^*$. Hence, the boundary value of $\boldsymbol{q}_{\mathrm{pe}}^*$ can simply be used as the boundary conditions multiplier, avoiding the cumbersome and error-prone implementation of long expressions as in [41]. Finally, it should also be noted that the first equation then directly gives the adjoint boundary condition as a source for the adjoint flux, which is the natural discrete adjoint boundary condition [100].

Therefore, it is analyzed now when the terms $f_{\nabla_\theta \boldsymbol{q}}^\top$, $f_{\nabla_r \boldsymbol{q}}^\top$, $\mathcal{S}_{\nabla_\theta \boldsymbol{q}}^\top$, and $\mathcal{S}_{\nabla_r \boldsymbol{q}}^\top$ are effectively present. The first term $f_{\nabla_\theta \boldsymbol{q}}^\top$ is related to the integrand of the objective $\widetilde{\mathcal{F}} = (Q_{\mathrm{o}} - Q_{d,p})^2$. However, there is some flexibility in the formulation of the target heat load $Q_{\mathrm{o}} = Q_{\mathrm{t}} + Q_{\mathrm{sr}}$. For example, Q_{t} can either be given by the boundary condition

$$Q_{\mathrm{t}} = \delta_{\mathrm{sh}} T n_{\mathrm{i}} u_\theta \mathbf{e}_\theta \cdot \boldsymbol{\nu},$$

or by the heat flux

$$\mathbf{Q} \cdot \boldsymbol{\nu} = \left(\frac{5}{2} \sum_{a=i,e} \mathbf{\Gamma}^a T - \kappa \nabla T \right) \cdot \boldsymbol{\nu}.$$

Although these two expressions yield, in principle, the same value, their derivatives do not. Moreover, Q_t does not contain spatial derivatives, where $\mathbf{Q}\cdot\boldsymbol{\nu}$ does. For the components $\mathcal{S}^\top_{\nabla_\theta \boldsymbol{q}}$ and $\mathcal{S}^\top_{\nabla_r \boldsymbol{q}}$, a similar argument can be raised. The only boundary condition source terms containing spatial derivatives are the flux terms $\mathbf{\Gamma}^\mathrm{n}$ and $\mathbf{Q}$, through the occurrence of ion fluxes $\mathbf{\Gamma}^\mathrm{i}$. Similarly, these spatial derivatives do not occur if the boundary flux $\mathbf{\Gamma}^\mathrm{i}\cdot\boldsymbol{\nu}$ is substituted by the source $\mathcal{S}_{n_\mathrm{i}}$ of the local boundary condition

$$\mathbf{\Gamma}^\mathrm{i}\cdot\boldsymbol{\nu} = \mathcal{S}_{n_\mathrm{i}} = S_{\mathrm{c},n_\mathrm{i}} + S_{\mathrm{v},n_\mathrm{i}} n_\mathrm{i}, \tag{5.26}$$

with $S_{\mathrm{c},n_\mathrm{i}}$ and $S_{\mathrm{v},n_\mathrm{i}}$ the constant and variable part of the boundary condition source as outlined in section 3.3.4.

These observations motivate the following pragmatical approach. Instead of deriving the multipliers $\boldsymbol{p}^*$ for the specific choice of boundary condition formulations in the code, the code is slightly altered. The boundary condition sources $\mathcal{S}_{n_\mathrm{i}}$ of the continuity equation are stored for use in the neutral and energy boundary conditions. Anywhere a flux occurs in the boundary condition source $\mathcal{S}$ of objective function integrand $\widetilde{\mathcal{F}}$, the occurrence is replaced by these sources. As such, the consistent boundary multipliers for the code implementation obey the equality $\boldsymbol{p}^* = \boldsymbol{q}^*_\mathrm{pe}$. The discrete sensitivity then becomes

$$\delta\hat{\mathcal{J}}_\mathcal{H} = \sum_i (\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{i}})^\top \left\{\widetilde{\boldsymbol{\mathcal{B}}}_{\boldsymbol{\varphi}}\right\}^i_\mathcal{H} \boldsymbol{\delta\varphi}\Omega^i_\mathrm{pe} + \sum_j \left(\left\{\widetilde{\mathcal{F}}_{\boldsymbol{\varphi}}\right\}^j_\mathcal{H} + (\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{j}})^\top \left\{\widetilde{\boldsymbol{\mathcal{C}}}_{\boldsymbol{\varphi}}\right\}^j_\mathcal{H}\right)\boldsymbol{\delta\varphi}\Sigma^j,$$

with $\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{j}}$ the values of the adjoint multipliers in the (very small) boundary cell. This sensitivity expression, which is consistent with the discrete code, can then be approximated by

$$\begin{aligned}\delta\hat{\mathcal{J}}_\mathcal{H} &\approx \sum_i (\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{i}})^\top \partial_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{B}}}_h \boldsymbol{\delta\varphi}\Omega^i_\mathrm{pe} + \sum_j \left(\partial_{\boldsymbol{\varphi}}\widetilde{\mathcal{F}}_h + (\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{j}})^\top \partial_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{C}}}_h\right)\boldsymbol{\delta\varphi}\Sigma^j,\\ &\approx \sum_i (\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{i}})^\top \Delta^\varepsilon_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{B}}}_h \Omega^i_\mathrm{pe} + \sum_j \left(\Delta^\varepsilon_{\boldsymbol{\varphi}}\widetilde{\mathcal{F}}_h + (\boldsymbol{q}^*_{\mathrm{pe},\mathcal{H},\boldsymbol{j}})^\top \Delta^\varepsilon_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{C}}}_h\right)\Sigma^j,\end{aligned}$$

where the finite difference evaluations $\Delta^\varepsilon_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{B}}}_h$, $\Delta^\varepsilon_{\boldsymbol{\varphi}}\widetilde{\mathcal{F}}_h$, and $\Delta^\varepsilon_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{\mathcal{C}}}_h$ can be evaluated simultaneously.

In the discrete code, finding a directional finite difference sensitivity then comprises evaluation of magnetic field and grid generator for the perturbed

control variables $\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}$ and using the perturbed grid generation variables $\boldsymbol{q}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}))$ throughout one iteration of the plasma solver to compute all residuals, without actually solving the correction equation and updating the state $\boldsymbol{q}_{\mathrm{pe}}$. It is important to realize that the cost of such a sensitivity evaluation is negligible with respect to the cost of a nonlinear iterative solution of the plasma edge equations. Because of this low computational cost, a central difference approximation will be used for the in-parts adjoint gradient calculations to improve accuracy.

An additional advantage that is offered by this in-parts adjoint approach is the low derivational effort that it requires to change the design variables. The differentiation with respect to other code parameters can be performed easily at a low computational cost, without any additional derivational efforts. The approach therefore offers a significant advantage in applications where sensitivities with respect to many different sort of variables are needed, such as parameter estimation and robust design applications. It also allows easy extension of the current magnetic field optimization to optimization of e.g. coil position or neutral pump speed.

5.5 Sensitivity verification with a grid refinement study

In order to verify the correct implementation of the in-parts adjoint approach, the sensitivities are compared to simple forward finite difference sensitivities

$$\delta\hat{\mathcal{J}} \approx \Delta^{\varepsilon}\hat{\mathcal{J}} := \frac{\hat{\mathcal{J}}(\boldsymbol{\varphi} + \varepsilon\boldsymbol{\delta\varphi}) - \hat{\mathcal{J}}(\boldsymbol{\varphi})}{\varepsilon}.$$

The in-parts adjoint sensitivities $\Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{\mathcal{L}}$ are evaluated using central difference calculations, because of the relatively low computational cost at which they can be evaluated. Since the sensitivity calculation is based on a continuous adjoint plasma edge simulation, a difference in discretization error is to be expected between the two sensitivities, in addition to the cancellation and truncation errors of the finite difference calculations themselves. However, these differences should

decrease with increasing grid accuracy. In principle, they can be made arbitrarily small by grid refinement.

A grid refinement study is therefore performed here for the initial sensitivity of the JET case presented in chapter 4. The sensitivities are subsequently evaluated on grids of sizes 100×24, 200×48, and 400×96. In each of these refinement steps, the finer grids are created such that each second cell boundary coincides with one of the rougher grid. Only using such consistent refinement steps, it can be guaranteed that the discretization errors reduce. A refinement study of the in-parts adjoint and finite difference sensitivities of the four coils is shown in figure 5.1. Remark that grid dimensions in this refinement study are kept small on purpose because of the strong increase of computational cost with grid refinement and the associated cost of the finite difference calculations.

In the left subfigure, both sensitivities are shown as a function of grid resolution. Remark that, similar to the cost function normalization, the design variables are normalized with a characteristic current size to come to sensitivity components of the order of unity. Significant discrepancies between both sensitivity approximations are found. By the relative size of the discrepancies, it may be assumed that for these grids the discretization error dominates. Indeed, observation learns that the error significantly reduces with grid refinement.

In the right subfigure, the decrease can be observed more clearly. There, the difference is shown in a log-scale as a function of grid resolution. It can be observed that especially when going from the 200×48 to the finest 400×96 grid, the discrepancies start decreasing with the same grid convergence order. Analysis shows this convergence order is slightly above one, as expected. Remark finally that the sensitivities in this study change significantly with grid refinement, a direct consequence of the rough discretization of the radial transport. It can be concluded that sufficiently fine radial discretization is needed for an accurate optimization. The 240×80 grid, which is considered very fine in the plasma edge community, is a compromise between accuracy and computational cost.

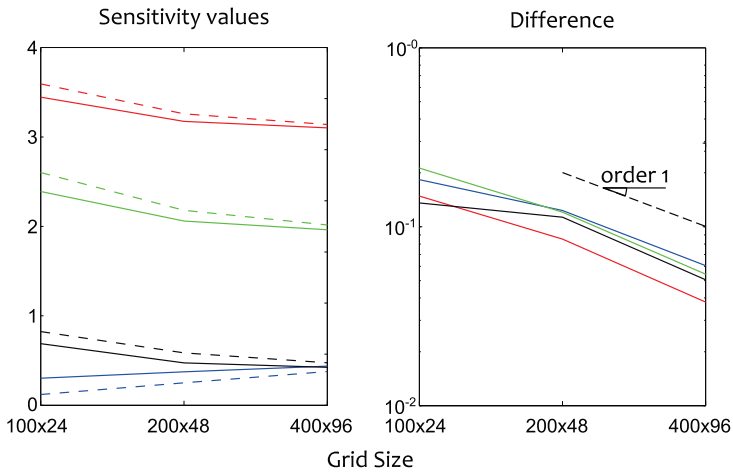


Figure 5.1: A grid refinement study of the in-parts adjoint approach and the finite difference approach to sensitivity calculation. The left subfigure shows a comparison of the normalized sensitivity components of $\nabla\hat{\mathcal{J}}$ approximated by the in-parts adjoint approximation (dashed lines) and a regular forward finite difference approximation (solid lines) as a function of grid resolution. On the right side, the discrepancy between the two approximations is shown along with a reference slope of first order decay. The different colours blue, red, yellow, and black correspond to the sensitivity contributions of coils 1, 2, 3, and 4 of figure 4.4, respectively.

5.6 Application to optimal current design

In this section, the power of the adjoint based procedure is illustrated in a slightly more challenging magnetic divertor design application. As explained earlier, the adjoint-based methodology for sensitivity calculation reduces the computational cost from approximately $n_{\boldsymbol{\varphi}}$ times the cost of a simulation to a fixed cost of about two simulations. Therefore, it allows increasing the number of considered design variables considerably. This is illustrated now by adapting the JET design case of chapter 4 (with $\lambda_P = 1$) to include more conductors (see figure 5.2).

A first application of the sensitivity approach could now be to estimate quickly where to best locate additional shaping coils, given the fifteen suggested locations

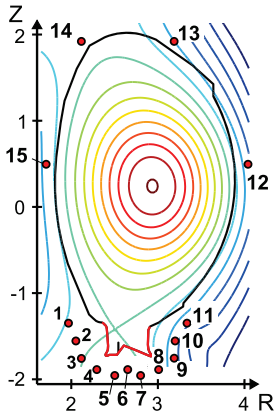


Figure 5.2: The set-up for magnetic field design. 15 new conductors (numbered) are placed around the vessel. The contour lines are magnetic flux lines. The target area protected by high-heat-flux plasma facing components (red line).

in figure 5.2. The sensitivity of the heat load cost functional (4.4) to all coil currents indicates which coils affect the target heat load most dominantly. For the initial flux function shown in figure 5.2, the conductors below the divertor show the biggest potential reduction from a linearized point of view, with a clear preference for conductor 6 (see figure 5.3). However, a sensitivity only contains local information (in design space) and for quantitative design purposes one at least needs information involving correlations between different control currents.

To obtain the combination of currents that mitigates best the target heat load, an iterative optimization approach, like the one outlined in chapter 4, is therefore needed. The same nested optimization approach is now applied, with the in-parts adjoint routine replacing the costly finite difference evaluations. Figure 5.4 shows the cost functional and its contributions at the different optimization iterations. From this figure, it is clear that the different terms in the cost functional are continuously balanced against each other throughout optimization. One can see for example that after 6 optimization steps the target heat load is already significantly lowered (thin solid line). Nevertheless, a significant heat load on the first wall is found (see intermediate heat load in figure 5.5). In the next steps, the

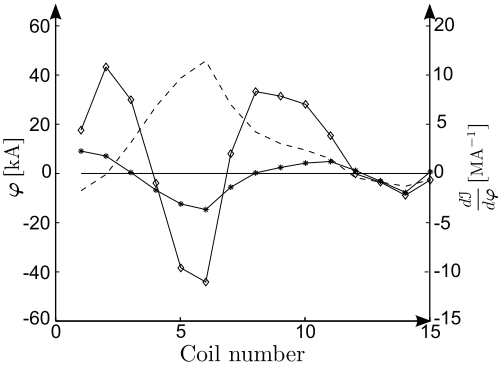


Figure 5.3: The evolution of shaping coil currents $\boldsymbol{\varphi}$ throughout optimization and the initial sensitivity $\dfrac{d\mathcal{J}}{d\boldsymbol{\varphi}}$. Initial values (—), values after 6 optimization steps (-∗-), final values (-◇-), initial sensitivity (– –)†.

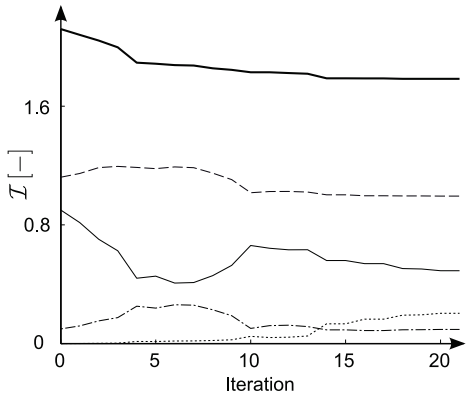


Figure 5.4: The evolution of the cost functional and its contributions during optimization. —: total cost functional, – – and — : respectively not simulated and simulated part of term (1) in Eq. (4.4), -·- : wall penalty (term (2) in Eq. (4.4)), ··· : control (current) penalty (term (3) in Eq. (4.4)).

wall heating term is decreased, while the target heat term is again increased.

†This figure was first published "Blommaert, M., Baelmans, M., Dekeyser, W., Gauger, N. & Reiter, D. (2015). A novel approach to magnetic divertor configuration design. *Journal of Nuclear Materials*, **463**, 1220–1224". Because of an inconsistency in the code

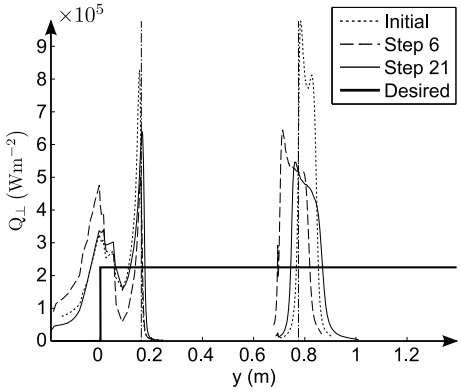


Figure 5.5: Target heat load before (···), during (after 6 steps, – –) and after (—) optimization compared to the desired heat load profile (—) as a function of a coordinate y along the target surface.

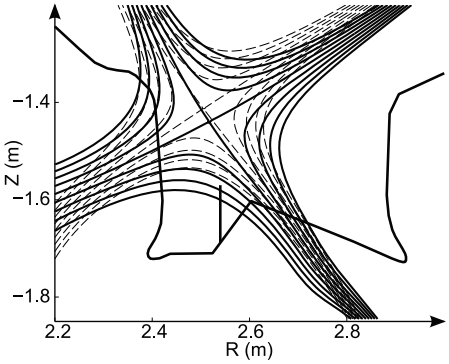


Figure 5.6: Poloidal magnetic flux before (– –) and after (—) optimization. Notice the change in flux tube expansion at the targets.

Eventually, a better compromise is found, according to the considered objective functional, where heat load and first wall heating are reduced at the price of increased current penalty/Joule losses. From figure 5.4, it is observed that in comparison to the intermediate design, outer target heat load and inner first wall

in the usage of the poloidal flux per radian ψ and total poloidal flux $\Psi = 2\pi\psi$, the currents $\boldsymbol{\varphi}$ were overestimated by a factor 2π. This is resolved here by rescaling the currents.

heat loads have been reduced significantly, while inner target heat load slightly increased. When studying the magnetic flux lines in figure 5.6, one clearly observes that field lines have widened towards the target area. Like in the previous chapter, the automated approach thus again evolves to a configuration that resembles the X-divertor proposed in Ref. [85].

It should be noted that the results might differ when more elaborate plasma edge models are used, although the same methodology for sensitivity studies can be employed. Use of a Monte Carlo code for neutral particle transport might, however, require careful analysis of the influence of noise on the overall procedure. Also in other optimization applications, this in-parts adjoint approach can be very valuable. A plasma edge application on which the author contributed is the estimation of plasma edge model parameters fitting to experiments [5]. The in-parts adjoint approach is of great value for this application, since often a multiplicity of parameters is estimated, such as parameters related to specific boundary conditions, coefficients in nonlinear transport coefficients, or even the coefficients in their spatial parametrizations as introduced in the master thesis of R. Dilissen [47]. Using the in-parts adjoint method, the inclusion of new parameters comes without derivational and implementational effort. The traditional continuous adjoint approach, in contrast, would require differentiating the equations by hand and implementing the expression in the code at the risk of introducing bugs. First steps have therefore been taken to include the in-parts adjoint approach also for parameter estimation problems [16].

5.7 Conclusions

In this chapter, an in-parts adjoint procedure has been established to speed up the design sensitivity calculations. It has been argued that the moving domain boundaries and some hidden source functions in the plasma grid generation hinder a full adjoint procedure. However, the low computational cost of magnetic field calculation and grid generation justify a procedure in which only the sensitivity of the plasma edge simulation is calculated using an adjoint methodology.

Such an in-parts adjoint methodology that includes the existing continuous adjoint plasma edge routine has therefore subsequently been derived. The re-

mainder of the sensitivity is then calculated using finite differences, avoiding all derivations. This, however, leads to a semi-discrete approach to gradient calculation, in which special attention is needed for the differentiation of the plasma edge boundary conditions. It has been found that by a slight reformulation of the boundary conditions, consistent adjoint multipliers for the approximate boundary condition sensitivities are trivially found.

These sensitivities were verified with full finite difference evaluations in a grid sensitivity study. As expected, the difference between the in-parts adjoint and finite difference sensitivities decreases monotonically as the grid is refined with a convergence order above one. To show the advantages of the adjoint method, the JET application of the last chapter has been repeated with an increased number of design variables. Finally, it is pointed out that once a single adjoint plasma edge simulation has been calculated, the in-parts adjoint method enables calculation of sensitivities with respect to model parameters or other design variables with great ease. Because of its high flexibility, it borrows itself for use in other applications such as parameter inference from experiments or robust design applications.

6

Integration of Free-Boundary Equilibrium Magnetic Model

In chapter 3, a first model has been introduced for magnetic divertor optimization. In a first stage, the magnetic field evaluations were based on a perturbation model that uses the vacuum approach for small divertor coil changes. Hereby, the influence is neglected of their respective magnetic field changes on the force balance in the main plasma. The validity of this model can therefore be questioned already in the magnetic field optimizations carried out in the former chapters, in which reasonably large optimal current changes have been found. Using this model, the design freedom is therefore limited to small adjustments of configurations that have already been assessed with more advanced magnetic equilibrium codes. Therefore, this chapter deals with the inclusion of a more accurate and consistent magnetic field model. As such, the design freedom of the new divertor optimization tool will be greatly enhanced.

As explained in section 3.1, the static magnetohydrodynamic equilibrium in tokamaks is governed by the Grad-Shafranov (GS) equation. However, many equilibrium codes exist that solve this equation with different objectives and functionalities. A large part of these codes are developed for reconstruction of magnetic equilibria from experimental data, such as EFIT. Other so-called *fixed-boundary codes* such as HELENA, can be used for accurate solution of the GS equation with higher order finite element methods. These codes, however, require that the plasma boundary is already known or count on the user to prescribe a

desired boundary geometry. In a second step, the feasibility and coil configuration then needs to be obtained using a *free-boundary equilibrium* (FBE) code in inverse mode. For the purposes in this thesis, an FBE solver in direct mode is needed. These direct calculations simulate the magnetic field resulting from the coil currents without presumptions on the poloidal magnetic flux or the plasma boundary.

In the latter category, the CEDRES++ equilibrium code is a suitable candidate, not least due to the fact that the nonlinear solver includes a full linearization of the finite element code [61, 71, 72]. This enables first of all a fast and stable Newton solver, which leads to robust convergence while other codes face convergence issues related to physics-based vertical instabilities [72]. As a second consequence, the discrete adjoint sensitivities of the magnetic equilibrium are readily available. If a full adjoint procedure for the divertor design comes within reach in the future, this will for sure be an additional asset. The recent coupling of CEDRES++ to the European transport solver (ETS) [32] and CHRONOS [2] transport codes additionally create possibilities for future model extensions. Relevant for the current MATLAB divertor optimization code is the existence of MATLAB version of CEDRES++, also referred to as 'FEEQS.m'. Finally, it is an asset that the code has been benchmarked against other equilibrium codes, see e.g. [71].

This chapter is structured as follows. In a first section, the CEDRES++ FBE code is briefly presented. Next, in section 6.2, a comparison is made between the perturbation model and the CEDRES++ code to verify consistency of the model outputs and analyse the discrepancies. As such, the legitimacy of the perturbation model is studied. Subsequently, the integration of the CEDRES++ code with grid generator and plasma edge model is considered. It is shown that the extreme sensitivity of the plasma edge grid generator on the flux function accuracy around the stationary points of the magnetic field necessitates adaptive refinement in the FBE code. Afterwards, first divertor optimization studies are shown for the new WEST (tungsten (W) Environment in Steady-state Tokamak) divertor, currently under construction in the Tore Supra tokamak at CEA (Commissariat à l'Energie Atomique, France) [24, 26]. The accuracy of the in-parts adjoint sensitivities is first studied with finite difference calculations. Finally, after choosing a set of

appropriate constraints for this new test case, the optimization algorithm is used to optimally spread the heat load over the divertor target geometry.†

6.1 CEDRES++ free-boundary equilibrium calculations

The CEDRES++ code is a free-boundary equilibrium code with many functionalities. A full overview is given in Ref. [72]. For steady-state applications, the FBE equations reduce to (3.12), which are repeated here for clarity:

$$L\psi = -\mathrm{R}\,\nabla\cdot\left(\frac{1}{\mu(\psi)\mathrm{R}^2}\nabla\psi\right) = J_\phi \qquad \text{, with}$$

$$J_\phi = \begin{cases} J_\mathrm{P} = \mathrm{R}p^{'}(\psi) + 1/(\mu_0\mathrm{R})\; FF^{'}(\psi) & \text{in } \Omega_\mathrm{mp}\,(\psi)\,, \\ I_{\phi,i}/S_i & \text{in } \Omega_{\mathrm{c},i}, i = 1\ldots n_c, \\ 0 & \text{elsewhere.} \end{cases}$$

In the core region, the Grad-Shafranov equation

$$L\psi = \mathrm{R}p^{'}(\psi) + 1/(\mu_0\mathrm{R})\; FF^{'}(\psi)$$

is retrieved that describes the balance between Lorentz forces and plasma pressure. In this expression, the flux functions p and F related to plasma pressure and toroidal magnetic field appear. In theory, p and F can be found by augmenting this equation with a core transport model, such as for example done in the Dina code [83]. Although recent projects aim at coupling CEDRES++ to core transport codes [2, 32], they are at present not stable and mature enough to incorporate in this optimization tool. Other disadvantages are the increased CPU-cost, the absence of a full Newton solver and the significant increase in complexity.

†Parts of this chapter have been published in "Blommaert, M., Heumann, H., Baelmans, M., Gauger, N.R. & Reiter, D. (2016). Towards Automated Magnetic Divertor Design for Optimal Heat Exhaust. *ESAIM: Proceedings and Surveys*, **53**, 49–63" and "Blommaert, M., Baelmans, M., Heumann, H., Marandet, Y., Bufferand, H. *et al.* (2016). Magnetic Field Models and their Application in Optimal Magnetic Divertor Design. *Contributions to Plasma Physics*, **56**, 796–801".

6. INTEGRATION OF FREE-BOUNDARY EQUILIBRIUM MAGNETIC MODEL

In the standard version of CEDRES++, it is assumed that the profiles $p^{'}$ and $FF^{'}$ are given in the form [18, 96]

$$p^{'}(\psi) = \lambda \frac{\beta}{\mathrm{R}_0}(1-\bar{\psi}^{\alpha})^{\gamma} \quad \text{and} \quad FF^{'}(\psi) = \lambda(1-\beta)\mu_0 \mathrm{R}_0(1-\bar{\psi}^{\alpha})^{\gamma}, \tag{6.1}$$

with $\bar{\psi}$ the normalized magnetic flux, which equals 0 at the magnetic axis and 1 at the last closed flux surface (LCFS). The physical meaning and choice of the profile parameters α, β, γ are dealt with in [18]. As explained there, a logical choice is to choose $\alpha = 1$, after which the β coefficient is set by the choice of poloidal beta (ratio of plasma pressure and magnetic pressure) and the γ coefficient is set by the internal magnetic inductance of the plasma. The remaining parameter λ can be determined as a scaling variable that is chosen as to match the total plasma current

$$I_{\mathrm{P}} = \lambda \int_{\Omega_{\mathrm{mp}}} \left[\left(\frac{\mathrm{R}}{\mathrm{R}_0}\beta + (1-\beta)\frac{\mathrm{R}_0}{\mathrm{R}} \right) (1-\bar{\psi}^{\alpha})^{\gamma} \right] d\omega$$

in a certain configuration[†].

The magnetic permeability is considered to be different from zero only in ferromagnetic structures, where μ has a strongly nonlinear dependence due to saturation effects. This is accounted for by linear interpolation from tables [72]. The code might both result in limiter or divertor configurations. In the former case, the boundary of the plasma domain Ω_{mp} touches an interior point of the reactor wall (limiter configuration). In the latter, the boundary contains one or more saddle points of ψ (divertor configuration). The equations are spatially discretized using a finite element (FE) method with linear Lagrangian elements on an unstructured triangular mesh (see figure 6.1). The discretized equation was linearized by hand and is solved for ψ using a Newton method (see Ref. [72]). It is clear that the nonlinear dependence of the plasma boundary Ω_p on ψ contributes to the complexity of this code [72].

[†]In the sensitivity calculations and optimization studies further in this chapter, the value of λ is calculated from a desired plasma current I_{P} for the initial configuration and then kept constant. From a control point of view, it could be interesting to keep the total plasma current I_{P} itself constant during optimization. However, for the simulations shown in this work, this option could not be activated yet in the FEEQS.m code. Recently, the code was extended with this functionality.

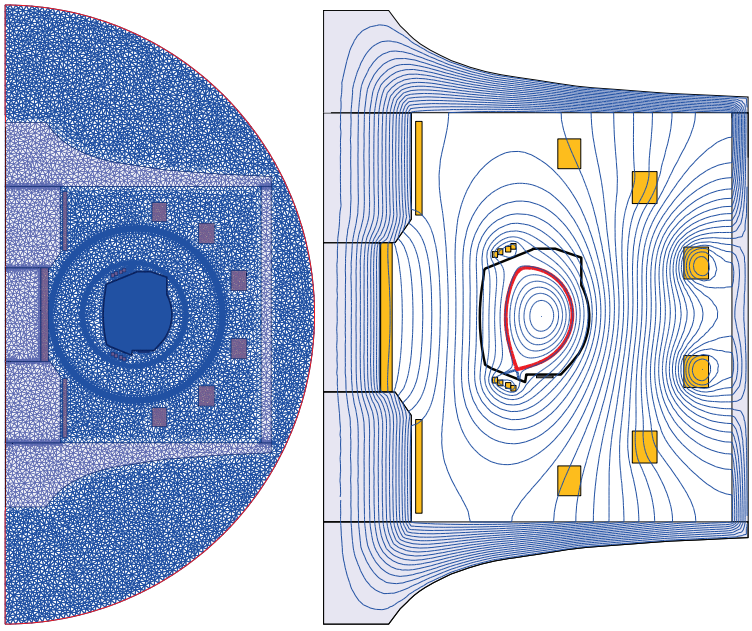


Figure 6.1: The triangular grid for the finite element simulation (left) and the magnetic flux function resulting from an FBE simulation (right) are depicted for the WEST reactor. In both figures, the iron structures are coloured light grey and the coils orange.

Finally, the free-boundary equilibrium equations need to be complemented with appropriate boundary conditions. At the symmetry axis, constant ψ is imposed. By convention, $\psi = 0$ is imposed here. The resulting effect of imposing such constant poloidal flux at the symmetry boundary is that no flux lines intersect with the boundary (see figure 6.1). In principle the other boundary condition would comprise imposing constant flux at infinity. However, grids can not be made infinitely large. In CEDRES++, this is solved by combining the finite element method with a boundary element method [1, 30]. The MATLAB version, on the contrary, does not include this feature and imposes conditions at a sufficiently far boundary. For the WEST case under study in the following sections, this is a reasonable approximation. Indeed, as can be seen in figure 6.1,

the presence of an iron transformer will function here as a de facto boundary condition that aligns all flux lines to the iron. As such, the exact location of the boundary condition will only marginally influence the magnetic flux within the vacuum vessel. For more details on the CEDRES++ code and its procedures the reader is referred to Refs. [19, 72].

6.2 Comparison of magnetic models

In order to compare the perturbation model (PM) presented in section 6.2 to the FBE model, the main assumptions specific to the perturbation approach are briefly repeated. These assumptions include:

1. the plasma current density changes δJ_ϕ can be neglected,
2. the magnetic flux change $\delta\psi$ can be calculated in vacuum. In addition to 1), this means that the permeabilities μ of surrounding materials are approximated by the vacuum permeability μ_0,
3. the coils can be approximated as having infinitesimal size.

6.2.1 Analysis of model assumptions

First, a consistency check is performed by verifying that the (MATLAB) CEDRES++ code reproduces the results of the perturbation model if similar approximations are introduced. As such, the correctness of implementation is verified and the effect of the different assumptions can be estimated. The WEST tokamak, for which the study in this paper is performed, is shown in figure 6.2 with coils, iron and the vacuum chamber clearly indicated. The divertor coils (numbered 10-17) will be of specific importance further on to control the magnetic configuration in the vacuum chamber. The reference magnetic equilibrium ψ_0 is obtained using the FBE code described in the previous section, with parameters $\alpha = 1$, $\beta = 1.5$, $\gamma = 0.9$, and $I_p = 0.47$ MA. The divertor coil currents of the reference configuration are found in table 6.1.

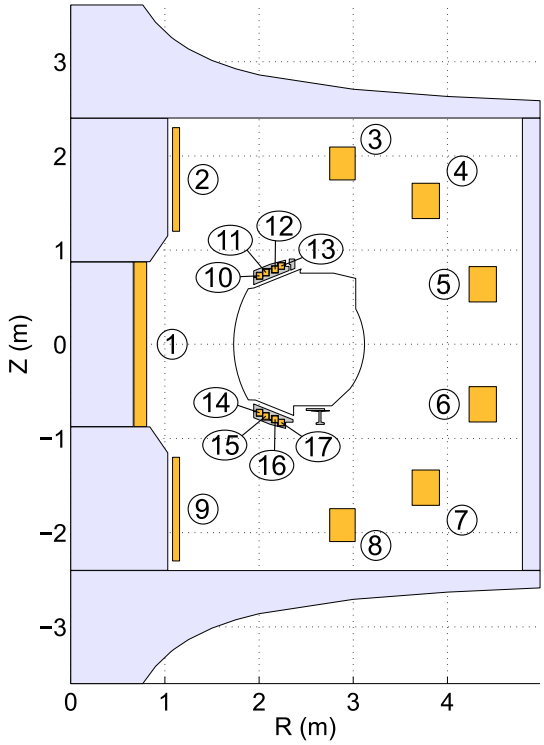


Figure 6.2: The components of the WEST machine, as they enter the FBE computation. The vacuum vessel is indicated by the black line, coils are indicated in orange and numbered, iron structures are indicated in blue and passive structures around the coils in grey.

Conductor	1	2	3	4	5	6	7	8	9	10
I (kA)	0	0	0	22.3	-186.4	-186.4	-22.3	0	0	41.9
Conductor	11	12	13	14	15	16	17			
I (kA)	41.9	41.9	41.9	51.2	51.2	51.2	51.2			

Table 6.1: Values of the coil currents used in the reference FBE simulation. Conductor numbers corresponding to the numbers in figure 6.2.

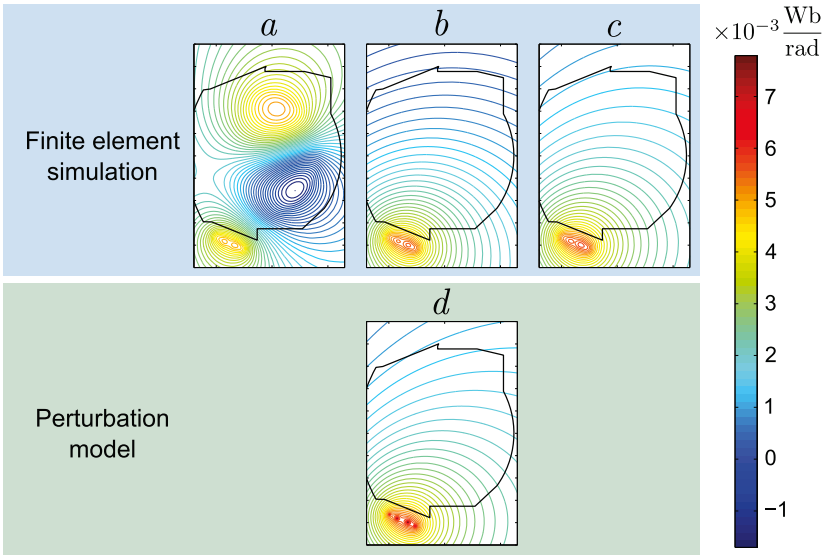


Figure 6.3: Close-up of the magnetic flux change $\delta\psi$ at a perturbation of 1kA of coils 14-17 at the vacuum vessel (black line) using: a) the FBE finite element simulation, b) the finite element simulation of the perturbation $\delta\psi$ with $\delta J_{\mathrm{P}} = 0$, c) the finite element simulation of $\delta\psi$ with $\delta J_{\mathrm{P}} = 0$ and neglecting permeability of iron, d) perturbation model (numerical evaluations of equations (3.19) and (3.20)).

The magnetic flux function $\psi_{1\mathrm{kA}}$ following from a simultaneous increase of all lower divertor coils (coil numbers 14-17) by 1 kA is now compared for the two magnetic field models. In figure 6.3, the change in flux function $\delta\psi = \psi_{1\mathrm{kA}} - \psi_0$ is shown for both the FBE model (figure 6.3a) and the PM (figure 6.3d). Figures 6.3b and c use the finite element code of the FBE simulation to simulate the same change in flux function, but introduce additional assumptions of the PM into the finite element code. In figure 6.3b the finite element code is used to simulate the flux change $\delta\psi$ in absence of plasma currents, while in figure 6.3c, all relative permeabilities are neglected in addition. The former is simply done by running the code with input parameters $I_{\mathrm{P}} = 0$ and loading the lower divertor coils exclusively with the change in current $\delta I_{\phi} = 1\mathrm{kA}$, so that the additional plasma

current density δJ_P is set to zero. The latter, in addition puts the nonlinear relative permeability function in the code to one.

Comparing the difference $\delta\psi$ from two full FBE simulations in figure 6.3a with that of the perturbation model in figure 6.3d, a significant difference is observed. Especially in the core region, a clear difference in magnetic flux function is present that can be attributed to the absence of the nonlinear interaction with plasma currents in the PM. This is confirmed in 6.3b, where $\delta\psi$ is shown for the finite element code without the effect of the plasma currents. Indeed, comparing figures 6.3b and d, a better agreement is now found. Additionally removing the relative permeability of iron from the CEDRES++ code finally gives an even better agreement of the FE code with the PM (figures 6.3c and d). Nevertheless, a remaining discrepancy can still be observed in the direct vicinity of the coils. This discrepancy can be attributed to the assumption of infinitesimally sized coils in the PM. The latter assumption, however, has almost no consequence on the magnetic field in the vacuum chamber. It only makes the peaks in flux function at the coil positions more pronounced.

6.2.2 Comparison of sensitivities

Next, we compare the first order behavior of the stationary points of ψ, i.e. the X-points and magnetic axis, for both models. These points are an important characteristic of the magnetic field. The aim is to estimate whether at least locally the models show similar tendencies and thus to analyse whether the perturbation model might suffice for sensitivity calculations.

The sensitivities $\partial\boldsymbol{p}/\partial\boldsymbol{\varphi}$ of the stationary point movement with respect to the coil current magnitudes is found by using a simple forward finite difference approach on both magnetic models. Here, $\boldsymbol{p} = \mathrm{R}\,\mathbf{e}_\mathrm{R} + \mathrm{Z}\,\mathbf{e}_\mathrm{Z}$ represents the position vector of the stationary point in the poloidal plane. A finite difference perturbation size $\varepsilon = \varepsilon_\mathrm{m}^{1/3}$ is used. It should be noted that the FE code solves the FBE equations for the poloidal flux ψ on an unstructured triangular grid with first-order elements. It is cumbersome in this unstructured grid to directly find the stationary point location $\boldsymbol{p}$, where the interpolated value of the derivative $\partial\boldsymbol{p}/\partial\boldsymbol{\varphi}$ is zero. Therefore, the values of the poloidal flux are first interpolated

to a fine rectangular mesh using natural interpolation, before searching for this location. In the MATLAB version of the CEDRES code, a routine is readily available that traces X-points based on the signs of flux differences between adjacent nodes. However, this routine is not suited to calculate the sensitivity of the X-point movement, since this approach leads to X-point positions that by definition coincide with grid nodes. The location of the X-point would therefore move in a discontinuous fashion when this routine would have been used.

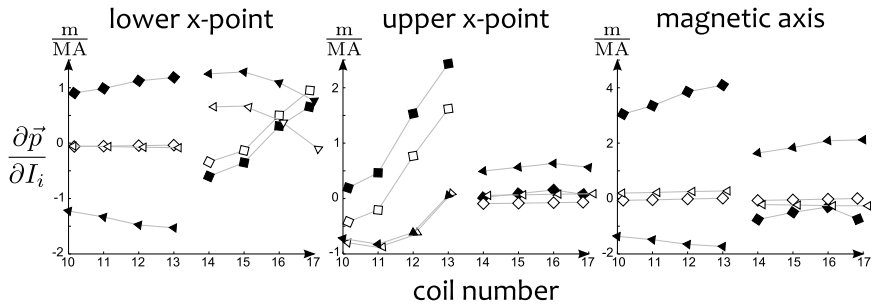


Figure 6.4: Comparison of the perturbation model and FBE simulation sensitivities. Sensitivities of lower and upper X-point, and magnetic axis positions with respect to the divertor coil currents are compared. The results of the finite element FBE simulation are shown with plain symbols, those of the perturbation model with empty symbols (R-component: squares, Z-component: triangles). Coil numbers are as given in figure 6.2.

The results of this study are shown in figure 6.4. Remark that overall, big discrepancies are found in the sensitivity analysis and that both models sometimes even have sensitivity contributions of opposite sign. However, reasonable agreement is found for the sensitivities of the lower X-point with respect to coils 14-17 and those of the upper X-point with respect to coils 10-13. This is a logical consequence of the fact that these coils are in the direct vicinity of these respective X-points under consideration. The magnetic field caused by a an infinitesimal piece of current conductor is inversely proportional to the squared distance from the current source, so that close enough to the coils the influence of the coil current dominates that of the plasma currents.

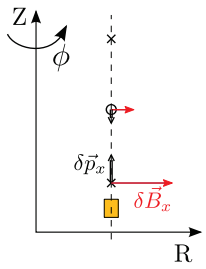


Figure 6.5: Simplified analysis of the stationary point movement in the vacuum approximation. The magnetic field change at the X-point due to a positive coil perturbation δI is shown in red and the corresponding movement of the X-point with a double arrow.

The discrepancies between the models are now illustrated with a simple example. Suppose that X-points, magnetic axis and a divertor coil are aligned as illustrated in figure 6.4. One can then easily study the linearized effect of plasma currents on the X-point location, or equivalently for the magnetic axis location. The displacement $\boldsymbol{\delta p} = [\delta p_R, \delta p_Z]^\top$ of a stationary point in poloidal flux ψ can be obtained by linearizing the spatial derivative of ψ around the stationary point location $\boldsymbol{p} = \boldsymbol{p}_0$ with respect to $\boldsymbol{\delta p}$. This reads

$$\frac{\partial \psi}{\partial \boldsymbol{p}} = \frac{\partial \psi}{\partial \boldsymbol{p}}\bigg|_{\boldsymbol{p}_0} + \frac{\partial^2 \psi}{\partial \boldsymbol{p}^2}\bigg|_{\boldsymbol{p}_0} \boldsymbol{\delta p} \qquad \text{and thus} \qquad \boldsymbol{\delta p} = -\left(\frac{\partial^2 \psi}{\partial \boldsymbol{p}^2}\bigg|_{\boldsymbol{p}_0}\right)^{-1} \frac{\partial \psi}{\partial \boldsymbol{p}}\bigg|_{\boldsymbol{p}_0}, \quad (6.2)$$

where the latter equality holds because $\partial\psi/\partial\boldsymbol{p} \triangleq 0$ at a stationary point of ψ. The X-point movement caused by a small change in coil current δI_ϕ (in positive $\hat{\phi}$ direction) is then given by

$$\frac{\partial \boldsymbol{p}}{\partial I} = -\left(\frac{\partial^2 \psi}{\partial \boldsymbol{p}^2}\bigg|_{\boldsymbol{p}_0}\right)^{-1} \frac{\partial^2 \psi}{\partial \boldsymbol{p}\, \partial I}\bigg|_{\boldsymbol{p}_0}, \text{with} \qquad \frac{\partial^2 \psi}{\partial \boldsymbol{p}\, \partial I} = \left[\mathrm{R}\frac{\partial B_Z}{\partial I}, -\mathrm{R}\frac{\partial B_R}{\partial I}\right]^\top \quad (6.3)$$

and where $\vec{B} = B_R\, \mathbf{e}_\mathrm{R} + B_\phi\, \mathbf{e}_\phi + B_Z\, \mathbf{e}_\mathrm{Z}$ is the magnetic flux density. The Hessian matrices with respect to R and Z, $H_x = \partial^2\psi/\partial\boldsymbol{p}^2|_{\boldsymbol{p}_x}$ at the X-points and $H_{ma} = \partial^2\psi/\partial\boldsymbol{p}^2|_{\boldsymbol{p}_{ma}}$ at the magnetic axis $\boldsymbol{p}_{ma}$, are in this case of the form $H_x = \mathrm{diag}\,(-a_1, a_2)$ and $H_{ma} = \mathrm{diag}\,(-a_3, -a_4)$, respectively, with coefficients

$a_1, a_2, a_3, a_4 \in \mathbb{R}^+$. The well-known right hand rule can be used to find the direction of $\partial\vec{B}/\partial I\,\delta I$ at the lower X-point caused by a positive perturbation δI[†]. Hence, equation (6.3) retrieves an upwards movement $\boldsymbol{\delta p}_x = \partial \boldsymbol{p}_x/\partial I\,\delta I$ of the lower X-point.

If one involves the plasma current I_P, the conclusion is less straightforward. Suppose the (toroidal) plasma currents can be represented using a thin toroidal wire at the magnetic axis location. The movement of the magnetic axis and therefore that of the plasma current can then be analyzed using equation (6.3). Due to the negative definite Hessian, one finds that the magnetic axis moves in the opposite direction of the X-point (downwards) and thus creates a counteracting magnetic field at the lower X-point. The direction of the X-point movement thus depends on the current I_P, the mutual distances of the variables and the coefficients a_2 and a_4. The resulting movement direction becomes less predictable because of this indirect effect.

The simplified analysis without plasma current should thus show us an approximate tendency for the sensitivities of the perturbation model. Looking again at figure 6.4, one sees this analysis indeed shows the right tendency for coils 14-17 of the lower X-point and for coils 10-13 at the upper X-point for the PM. One also sees that all other sensitivities using the PM are relatively close to zero because of the large distance between coil and stationary point. One may thus conclude that in those cases, the indirect effect of the plasma currents dominates, since the plasma currents are closer to those stationary points then the perturbed coil.

Finally, it appears from figure 6.4 that the (direction of the) movement of the magnetic axis in the FBE model is not correctly predicted by the perturbation model, so that the reduction of the plasma currents to a coil at the magnetic axis will not suffice to improve the model. Indeed, it is rather the whole change of the plasma core shape that effectively determines in which direction the plasma currents move. Although further examination of alternative perturbative models with preprocessed determination of the plasma current "gravity center" is possible, it may be concluded at this point that the results of the perturbation model differ significantly of those of the FBE model. It is therefore decided to incorporate the FBE model into the optimal design approach.

[†]The right hand rule follows directly from Ampères law.

6.3 Adapting the FBE calculations for automated target heat load evaluations

The generation of a structured grid for the discretization of the plasma edge transport equations $\boldsymbol{c}_{\mathrm{pe}}(\boldsymbol{q}_{\mathrm{gg}}, \boldsymbol{q}_{\mathrm{pe}}) = 0$ is a very delicate procedure that requires accurate approximations of poloidal and radial particle, momentum, and energy fluxes. In this context it is crucial to locate accurately the X-point position. Yet, when constructing a plasma edge grid, it is exactly this region around the X-point that shows the most irregularities. Present-day plasma edge grid generators therefore rely heavily on user-interaction to indicate the magnetic configuration (e.g. CARRE [97]) and post-process sensitive regions such as the region around X-points. Because of the automated character of the optimal design framework, these issues need to be avoided at all cost.

In section 3.2, several measures were already discussed to automate the grid generator. However, due to the specific synergy of the FBE code presented in section 6.1 and the grid generator, additional steps are necessary when combining them. In the first part of this section, it will be explained why these regions around the stationary points show these irregularities. To resolve this, an adaptive mesh refinement procedure has been introduced for the FBE code in collaboration with H. Heumann [15]. The procedure is able to reduce the mesh distortion to a desirable level, while keeping the memory consumption of the FBE solution acceptable.

6.3.1 Estimation of contour line errors

When constructing a grid aligned to the magnetic field, contour lines of the flux function are traced. Two sides of each grid cell will then coincide with these contour lines. Since strong radial refinement of these grids towards the separatrix is needed, it is needless to say that these contour lines need accurate tracing. However, first grid generation trials on CEDRES++ grids showed rough distortions around the X-point. To find out why, the position error on contour lines is analysed here.

It is clear that the positioning error of the contour lines exists of several components, such as the discretization error of the FBE solution and the residual error on the convergence of the nonlinear FBE equilibrium problem. However, only the interpolation error following from the linear interpolation that is used in the (standard MATLAB) contouring algorithm will be considered here. It will become clear that it is this error that blows up near regions of stationary points.

Let ψ_1 and ψ_2 be two values of the discretized poloidal flux ψ_h resulting directly from the solution of the discretized FBE and let Δx be the distance between their two locations x_1 and x_2 (see figure 6.6). Suppose a contour line with level value ψ_c is traced. The position $\alpha\Delta x$ $(0 < \alpha < 1)$ of the contour line with level ψ_c $(\psi_1 < \psi_c < \psi_2)$ is found by the contouring algorithm using linear interpolation.

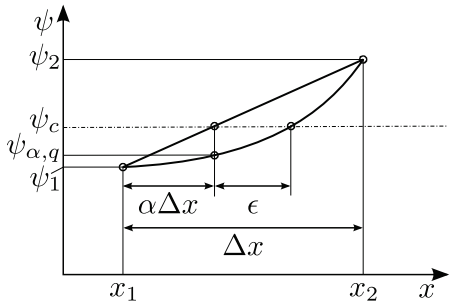


Figure 6.6: Error estimation on the contour line position when using linear interpolation based on a quadratic interpolation. The dash-dot line represents the contour level.

The error can then easily be estimated using a higher order interpolation. The derivation of the error starts by taking the Taylor expansion of ψ around ψ_1 to find ψ at an undefined distance Δx_1,

$$\psi = \psi_1 + \psi_1'(\Delta x_1) + \frac{1}{2}\psi_1''(\Delta x_1)^2 + \ldots, \tag{6.4}$$

where spatial derivatives are denoted with accents. Using only the linear part of the expansion at a distance $\alpha\Delta x$ and Δx, leads to the expression for the fraction

α following from linear interpolation

$$\alpha = \frac{\psi_c - \psi_1}{\psi_2 - \psi_1} \tag{6.5}$$

The error ϵ on the interpolated location can now be estimated by comparing to a quadratic interpolation. The expressions governing the quadratic interpolation can be easily found by including the quadratic terms of the Taylor expansion in Eq. (6.4) for points at distances of $\alpha\Delta x + \epsilon$ and Δx from the expansion center x_1 (see again figure 6.6). From these four relations, the error ϵ can then be eliminated for small values of ϵ (neglecting ϵ^2 terms), which gives

$$\epsilon = \frac{(\alpha - \alpha^2)\,\psi_1''\Delta x}{2\psi_{1,qc}' + 2\alpha\psi_1''\Delta x}, \tag{6.6}$$

where the subscript $(\cdot)_{qc}$ refers to the quadratic approximation. Next, using the relation $\psi_{\alpha,qc}' = \psi_{1,qc}' + \alpha\psi_1''\Delta x$, the error becomes

$$\epsilon = \frac{(\alpha - \alpha^2)\,\psi_1''\Delta x}{2\psi_{\alpha,qc}'}. \tag{6.7}$$

From equation 6.7, it can be clearly seen that the error in the contouring algorithm significantly increases towards extrema in ψ, i.e. when $\psi_{\alpha,qc}'$ is approximately zero. Simultaneously, it can be seen that the distance Δx between the solution points can be decreased accordingly in order to keep sufficient accuracy of the contour lines. This motivates the adaptive refinement procedure towards X-points introduced next.

6.3.2 Introducing adaptive grid refinement

The refinement of the unstructured CEDRES++ mesh is based on the longest edge bisection [7]. After each converged Newton iteration all triangles close to the X-points are marked for refinement and longest edge bisection is applied to generate a new triangular mesh with refined elements at the marked locations. The Newton method is next restarted to solve the FBE problem on this new mesh. The initial guess in the Newton iteration can be easily interpolated from the previous numerical solution on the coarser mesh and hence the Newton method converges in a couple of iterations.

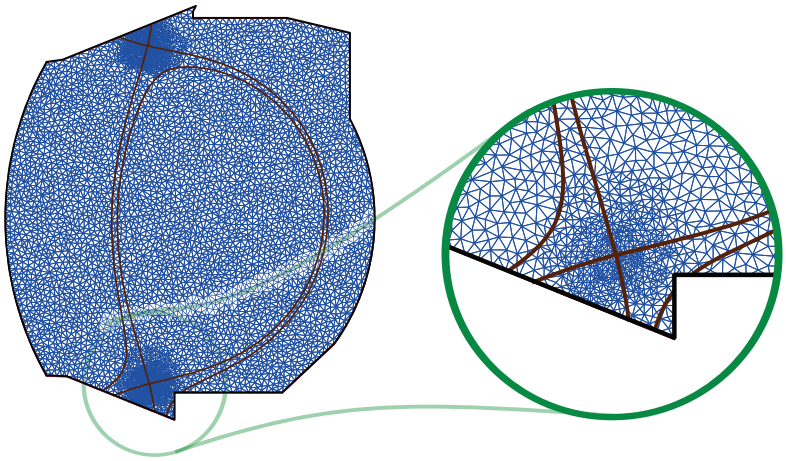


Figure 6.7: Fragment of the grid for FBE calculations within the vacuum vessel. Notice the adaptive refinement with respect to the two X-points (magnetic flux lines shown in brown).

Afterwards, more refinement steps are made until the desired accuracy is obtained in the X-point region. To assure that the grid changes sufficiently smoothly from rough to fine grid cells, the cells marked for refinement are based on the distance in grid cells from the X-point. By keeping this distance in grid cells constant, an ever smaller refined region will result from these grid adaptation steps. For the WEST case under consideration, 5 refinement steps are made towards the X-points. The result can be seen in figure 6.7. When optimizing, it suffices to perform only the last 2 or 3 steps adaptively within the magnetic flux calculation. The first grid adaptation steps are then simply performed for the initial configuration.

6.4 Sensitivity calculation and verification

With the adaptive refinement procedure for CEDRES++ at hand, the FBE code is integrated into the automated design framework. Sensitivity calculation and optimal design of the new model are now envisaged. This section is structured

as follows. In section 6.4.1, a topical WEST case will first be introduced. The WEST reactor is currently under construction in the Tore Supra tokamak at CEA (Commissariat à l'Energie Atomique, France) and is planned to start operation in the first half of 2016 [24, 26]. It will function as a divertor test platform in preparation for ITER and is as such equipped with an actively cooled tungsten divertor and novel divertor coils. Since operation scenarios for WEST are currently being planned, it is an attractive setting to test and demonstrate the techniques for magnetic divertor optimization. In section 6.4.2, the in-parts adjoint sensitivity calculations will be performed and verified with finite difference calculations. As such, it can be estimated whether the chosen plasma edge grid is sufficiently fine for accurate in-parts adjoint sensitivities of the novel WEST case. Subsequently, design constraints are set up in section 6.4.3 to finally study optimal design of the WEST configuration in section 6.4.4.

6.4.1 Test case description

Given the limitations of the plasma edge transport model, a test case similar to a SOLEDGE2D-EIRENE simulation discribed in Ref. [28] is chosen (the so-called high power FAR configuration, with a puff of $1.1 \cdot 10^{21}$ #/s). The set-up of the reference magnetic configuration was the one already described in section 6.2. The plasma edge simulation is performed on a 210×80 grid, as given in red in figure 6.8.

Two important differences in the plasma edge model with respect to the SOLEDGE2D-EIRENE simulation must be noted. At the one hand, a kinetic code for neutrals is absent in our model and replaced by the fluid neutral model. Therefore, additional assumptions on boundary conditions and neutral fluid properties need to be made. On the other hand, a methodology to simulate up to the reactor wall, such as in [4] or [27], is absent in the radial (perpendicular to the magnetic flux lines) direction. It can be seen in figure 6.8 that for the very open WEST configuration, only a fraction of the reactor is effectively simulated. While this is not expected to be a very crude approximation for the field aligned plasma, the limited grid size might give a very crude approximation of the neutral

transport. Therefore, extension to a simulation reaching up to the vessel wall is planned in future research [4, 27].

constant	value	units
D^{i}	0.6	$\mathrm{m}^2.\mathrm{s}^{-1}$
χ^{i}	2	$\mathrm{m}^2.\mathrm{s}^{-1}$
χ^{e}	2	$\mathrm{m}^2.\mathrm{s}^{-1}$
χ^{n}	0.2	$\mathrm{m}^2.\mathrm{s}^{-1}$
ν^{i}	0.2	$\mathrm{m}^2.\mathrm{s}^{-1}$
λ_n	0.05	-
λ_T	0.3	-
n_{c}	$1.3 \cdot 10^{19}$	m^{-3}
Q_{c}	7.93	MW
L	6.59	$\mathrm{m}^3\mathrm{s}^{-1}$
α_{c}	0.167	-
R_c	1	-
δ^i_{sh}	2.5	-
δ^e_{sh}	2	-
δ^{pot}_{sh}	2.5	-

Table 6.2: The values of transport and boundary condition parameters used in the WEST case.

The model parameters are given in table 6.2. The proportionality constant α_{c} for the neutral flux to the core is chosen to match the total neutral flux to the core region in the SOLEDGE2D-EIRENE simulation. Similarly, a smooth variable sticking fraction is used along the radial boundaries, so that the total pumped flux equals $7 \cdot 10^{20}$ #/s, matching the SOLEDGE2D-EIRENE pumped flux. The resulting total pumping speed of $6.59\,\mathrm{m}^3\mathrm{s}^{-1}$ is maintained through all simulations.

6.4.2 Sensitivity calculation and results

Even including the adaptive refinement, the time for a CEDRES++ code evaluation is in the order of ten seconds. This is still a negligible cost in comparison

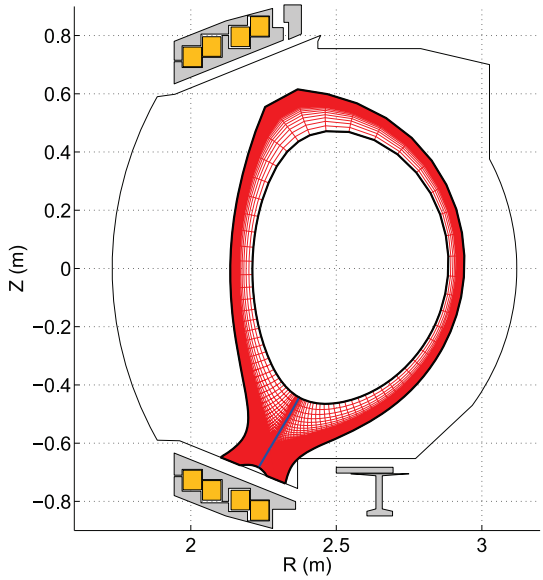


Figure 6.8: The grid for plasma edge calculations in the WEST vacuum chamber. The boundaries are indicated by a tick black line. The blue lines indicate the grid cuts that are used to unfold the grid onto a topologically rectangular computational mesh.

to the plasma edge transport simulation, that already takes a couple of hours for one code evaluation with the current physics model. Additionally, remark that the derivations in chapter 5 were carried out for a general nonlinear function $c_{\mathrm{eq}}(q_{\mathrm{eq}}, \boldsymbol{q}_{\mathrm{gg}}) = 0$. The in-parts adjoint approach can thus be applied. Although an adjoint procedure is at hand for the CEDRES++ part, it is inefficient to use adjoint sensitivity calculations for the FBE model as long as the grid generator code needs forward sensitivity calculations [11]. Indeed, the number of adjoint FBE solutions will then scale with the number of components of q_{eq}.

Again, a central difference approximation is used for the in-parts adjoint sensitivity

$$\Delta^{\varepsilon}_{\boldsymbol{\varphi}} \widetilde{\mathcal{L}} := \frac{\widetilde{\mathcal{L}}(\boldsymbol{\varphi} + \varepsilon \boldsymbol{\delta\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}}) - \widetilde{\mathcal{L}}(\boldsymbol{\varphi} - \varepsilon \boldsymbol{\delta\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}})}{2\varepsilon},$$

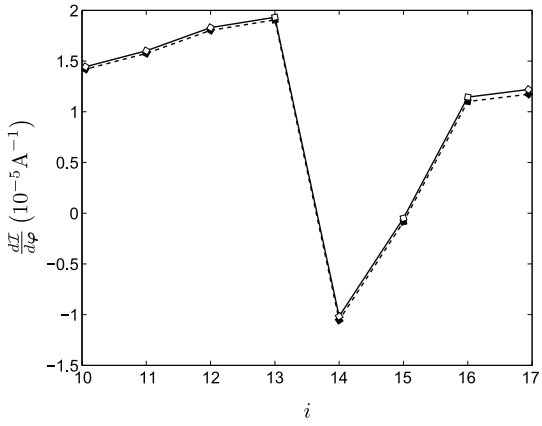


Figure 6.9: Evaluation of the design gradient $\nabla\hat{\mathcal{J}} = \left[\frac{d\mathcal{J}}{dI_{10}}\ \frac{d\mathcal{J}}{dI_{11}} \dots \frac{d\mathcal{J}}{dI_{17}}\right]^T$ using: the approximations $\Delta^{\varepsilon}\hat{\mathcal{J}}$ (thick dashed line) and $\Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{\mathcal{L}}$ (thick solid line) with perturbation size $\varepsilon = \varepsilon_{\mathrm{m}}^{\frac{1}{4}}$.

after solving the adjoint plasma edge transport equations. Since no grid sensitivity analysis will be made here, a central difference approximation can also be afforded for the full finite difference evaluation

$$\Delta^{\varepsilon}\hat{\mathcal{J}} := \frac{\hat{\mathcal{J}}\left(\boldsymbol{\varphi} + \varepsilon\ \boldsymbol{\delta\varphi}\right) - \hat{\mathcal{J}}\left(\boldsymbol{\varphi} - \varepsilon\ \boldsymbol{\delta\varphi}\right)}{2\varepsilon}.$$

The design vector $\boldsymbol{\varphi}$ for which the gradient of the objective functional is verified, consists of the shaping coil currents, i.e. $\boldsymbol{\varphi} = [I_{10}\ I_{11}\ \dots\ I_{17}]^T$. The comparison of the two gradient calculations is given in figure 6.9.

As can be seen from this figure, the partially adjoint gradient is reasonably accurate and remains close to the "full" central difference calculations, although it slightly overestimates the value of the different gradient components. Especially for the first four components it can be seen that there is a systematic (but small) deviation. This deviation can be mainly attributed to the discretization error corresponding to the use of a continuous adjoint procedure.

In table 6.3 the discrepancy

$$Err(\varepsilon) = \left|\Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{\mathcal{L}} - \Delta^{\varepsilon}\hat{\mathcal{J}}\right|,$$

of the two derivative approximations $\Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{\mathcal{L}}$ and $\Delta^{\varepsilon}\hat{\mathcal{J}}$ is provided for different finite difference step sizes ε. One can see that if ε is decreased from $\varepsilon = \varepsilon_m^{\frac{1}{4}}$ to $\varepsilon = \varepsilon_m^{\frac{1}{5}}$, the discrepancy decreases. Remark that apart from the cancellation and the truncation error, also the discretization error is influenced by the step ε. Of course, each change in control variable causes a changed plasma edge grid. Therefore, a typical balance between truncation and cancellation errors as in many finite difference step studies is not necessarily to be expected here, as one can observe in table 6.3. This confirms that the discretization error represents a prominent part of the discrepancy.

ε	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}
$\varepsilon_{\mathrm{m}}^{\frac{1}{3}}$	0.0235	0.0254	0.0271	0.0267	0.0242	0.0252	0.0178	0.0094
$\varepsilon_{\mathrm{m}}^{\frac{1}{4}}$	0.0236	0.0254	0.0271	0.0267	0.0376	0.0364	0.0431	0.0471
$\varepsilon_{\mathrm{m}}^{\frac{1}{5}}$	0.0120	0.0136	0.0150	0.0144	0.0131	0.0171	0.0138	0.0130

Table 6.3: The absolute difference of the finite difference and in-parts adjoint evaluations of the design gradient components $\delta\hat{\mathcal{J}}$, using the central difference approximations $\Delta^{\varepsilon}\hat{\mathcal{J}}$ and $\Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{\mathcal{L}}$, respectively. The difference is shown as a function of the finite difference perturbation size ε, which is expressed relative to machine precision ε_{m}.

As predicted, the plasma edge simulation dominated the computational cost. The gain of the in-parts adjoint approach with respect to the central difference calculation was thus roughly the predicted factor $\big(2 \cdot n_{\boldsymbol{\varphi}} + 1\big)/2 \approx n_{\boldsymbol{\varphi}} = 8$. However, the gain slightly decreased for smaller finite difference step sizes. In this case, the perturbed plasma edge simulations can benefit more from using the reference simulation as a good initial solution.

6.4.3 Setting up design constraints

Before optimal design can be initiated, appropriate design constraints have to be considered for the WEST magnetic divertor design problem. In practice, only very few design freedom is foreseen in the WEST reactor since only two power supplies are currently present. One is foreseen for the lower and one for the upper divertor

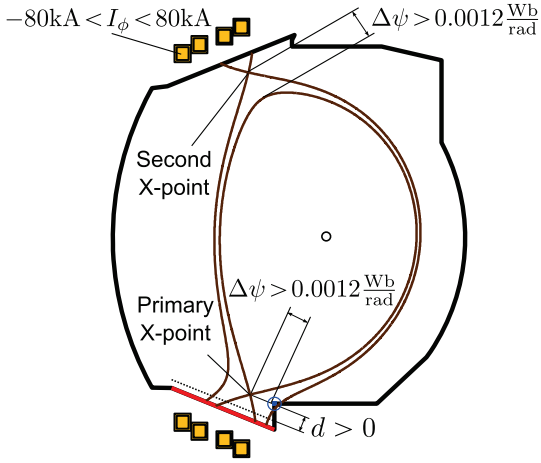


Figure 6.10: An overview of the magnetic configuration constraints in the WEST case. The brown lines represent the magnetic field lines of the primary and secondary X-point. The blue circle accentuates the first point that touches the magnetic field lines and thus determines the outer grid boundary.

coils. The coils are then connected in series. However, if worthwhile, investments in more current sources can always be made. To illustrate the potential of the method, it will be assumed that each coil has a separate bi-directional current source with a limit of 80 kA in a first hypothetical optimization study. This will lead to a set of 16 box constraints.

In addition, nonlinear constraints on the magnetic configuration are introduced. To guarantee a divertor configuration is kept, the X-point is forced to keep an orthogonal distance of 3.1 cm with respect to the target surface. Or equivalently, the X-point has to stay above the dotted line in figure 6.10. Finally, in order to be able to generate grids at all times, the scrape-off layer (SOL) width is constrained to a minimal poloidal flux span of $1.2 \cdot 10^{-3}\frac{\text{Wb}}{\text{rad}}$. As discussed in chapter 3, the computational grid in the SOL extends to the outermost flux line that runs from plate to plate without touching other vessel parts. However, since the MATLAB plasma edge code is not adapted for simulation of configurations with multiple X-points, the SOL grid boundary is reduced to the flux surface of the second X-point if necessary.

It has been pointed out in chapter 4 that for an SQP optimization, the linearization of these constraints with respect to the coil currents is needed. To avoid discontinuities in the constraint sensitivities, the minimal SOL width enters as two separate constraints (flux span between the X-points enters as a separate constraint). An overview of the constraints is given in figure 6.10. To calculate the gradient of the constraints with respect to the coil currents $\nabla\widehat{\boldsymbol{h}}$, finite difference evaluations are used. Notice that adjoint evaluations are not very beneficial here since the constraint vector $\boldsymbol{h}$ consists of at least three components here. Moreover, the constraint derivatives include i.a. the derivative of the X-point location. As can be seen from Eq. (6.3), second order spatial information of ψ is needed for an analytical description of the X-point movement, while the finite element FBE simulation leads to unstructured piecewise linear data. Since the constraints only involve magnetic field evaluations, the gradient cost is finally not of great influence.

6.4.4 Optimal design results

The optimization algorithm including FBE calculations is now used to find an improved magnetic configuration for WEST by designing the coil currents in coils 10-17 (see figure 6.2a). The optimization algorithm was stopped when it settled after five optimization cycles. Moreover, 92% of the changes in cost functional was realized in the first two cycles. The entire optimization came at the cost of roughly twelve plasma edge simulation equivalents, with free-boundary equilibrium calculation and grid generation cost negligible to that of a plasma edge simulation. Remark that this can be easily reduced to six plasma edge simulation equivalents for practical design purposes. Indeed, given the model accuracy, one might question the relevancy of the final 8% cost function changes.

In figure 6.11 a and b, the heat load and magnetic field change resulting from the optimization are given, respectively. It is observed in figure 6.11a that the peak heat load is reduced by 56%. From figure 6.11b it can be seen that this is realized by a significantly enlarged projected area through flux expansion. In comparison to the JET optimization in chapter 4, it is even more visible here that the X-point is drawn towards the target surface and limited eventually by

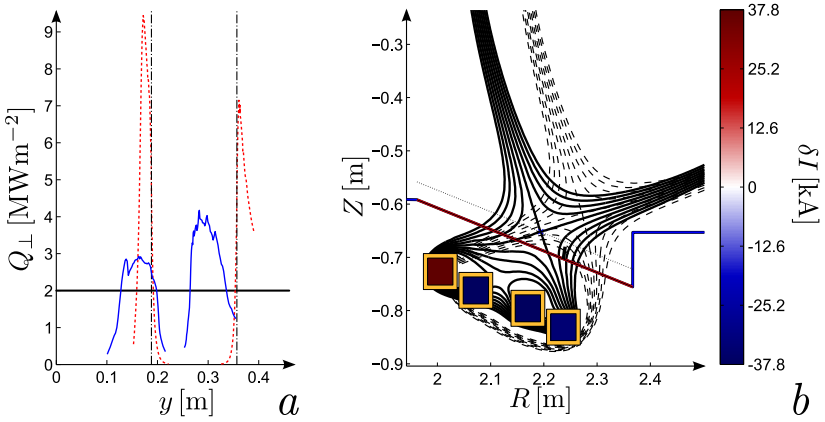


Figure 6.11: a) Heat load profile on the target surface in comparison to the desired profile Q_d, before (dotted) and after optimization (solid line). b) Close-up of the magnetic configuration change at the divertor region from initial (dashed) to optimized (solid). The colorbar indicates the change in current magnitudes δI. The dotted line represents the design constraint.

the constraint. Indeed, since the X-point is a magnetic flux null, the poloidal flux expansion reaches a maximum here.

The flux expansion effect on the heat load clearly dominates the effect of the reduced connection length in the current plasma regime. Note that a detached configuration might hold a different balance between these two means to spread the heat flux. In the detached regime, the enhanced radial heat convection due to the recycling processes might diffuse the heat better over the target surface and therefore favour a long connection length. Similar studies should therefore be repeated for the more interesting but more computationally intensive detached regime. It should be noted that the introduction of a constraint on the minimal distance between X-point and target was essential to avoid the optimization algorithm from retrieving a near-limiter configuration in which excessive quantities of impurities might find their way to the plasma core. This would namely lead to increased core impurity radiation and as a consequence deteriorated energy confinement.

Another magnetic configuration that combines enhanced connection length with increased flux expansion is the snowflake divertor configuration [122, 124]. However, given the current coil configuration, it is not expected that a snowflake divertor can be reproduced in WEST for reasonable ranges of the coil currents. Moreover, the current code is not capable of performing simulations for this type of configurations. Code extensions to study optimization of snowflake configurations have therefore been initiated recently [144].

Apart from moving the X-point, also the increase and decrease of neighbouring coils 14 and 15, respectively, realizes an additional flux expansion. These results are in agreement with the results of the JET optimal design study in chapter 4, since a clear similarity with the X-divertor concept is again found [85]. Finally, it should be noted that the total coil current was reduced by 94 kA. The optimization algorithm was thus able to produce a better configuration while reducing the Joule losses.

model	perturbation	FBE
$\Delta\hat{\mathcal{J}}\big/\hat{\mathcal{J}}$ (%)	-36.8	-55.8
$\bar{I}_{10}$ (kA)	63.6	32.4
$\bar{I}_{11}$ (kA)	53.4	30.1
$\bar{I}_{12}$ (kA)	25.3	28.6
$\bar{I}_{13}$ (kA)	-11.0	32.7
$\bar{I}_{14}$ (kA)	80.0	75.3
$\bar{I}_{15}$ (kA)	18.7	13.4
$\bar{I}_{16}$ (kA)	31.8	31.6
$\bar{I}_{17}$ (kA)	28.4	34.3

Table 6.4: Comparison of optimal configurations achieved with perturbation and FBE model. The relative change in cost function $\Delta\hat{\mathcal{J}}\big/\hat{\mathcal{J}}$ and optimal coil currents are shown for both the optimization studies with the perturbation model and the FBE model. Subscripted conductor numbers corresponding to the numbers in figure 6.2.

The optimization procedure was also repeated with the perturbation model. A similar tendency was found there. Again, the movement of the X-point towards

the target accompanied flux expansion and a decreased heat load. Nevertheless, the peak heat load was only reduced by 34 % in the optimal configuration. To explain this difference, the optimal coil currents and relative change in cost function are compared for both magnetic field models in table 6.4. It is observed that the perturbation model shows a lower relative decrease in cost function $\Delta\hat{\mathcal{J}}\big/\hat{\mathcal{J}}$ from initial to optimum. This indeed reflects the smaller decrease in peak heat load. The cause may be found in the values of the coil currents. Changes in currents have to be significantly larger in the perturbation model to realize a similar change in magnetic configuration (currents are initialized at 41.9 kA for coils 10-13 and 51.2 kA for coils 14-17, as listed in table 6.1). This in turn causes conductor 14 to reach its maximal value of 80 kA during optimization, impeding further improvement. It can be concluded that by ignoring plasma currents, the perturbation model significantly underestimates the effect coil currents can have on the magnetic configuration.

6.5 Conclusions

In this chapter, the perturbation model (PM) that uses the vacuum approach is evaluated and compared to a more complete free-boundary equilibrium (FBE) model. In a comparison of both models, it is found that an upgrade to an FBE model is indispensable for magnetic divertor design. Therefore, a MATLAB version of the CEDRES++ finite element FBE code is integrated into the code framework. In an automated combination with the plasma edge code, the standard grid accuracy of CEDRES++ in the X-point regions is found to be insufficient. By introducing an adaptive refinement strategy, smooth mesh adjustment towards these X-points is achieved.

Next, a topical WEST case is introduced in order to demonstrate the functionalities of the improved code. After setting up plasma edge grid and plasma edge model for the new case, sensitivities of the cost functional are shown to hold reasonably good accuracy. Finally, an optimal design was retrieved with a CPU-cost of about 12 times that of a plasma edge simulation. With a peak heat load reduction of 56% and a coil current reduction of 94kA, the optimal design strategy again shows its virtues. The obtained results are very plausible. No

doubt, however, that further improvements to the neutral model would increase the relevancy of the design tool for the very open WEST configuration.

6. INTEGRATION OF FREE-BOUNDARY EQUILIBRIUM MAGNETIC MODEL

7

One-shot Optimization

The nested optimization approach elaborated in the previous chapters decouples the optimization algorithm from the solution of the state and adjoint equations. The latter only serve as tools for objective function and gradient evaluation.

This has several advantages. First of all, the approach allows leaving the forward solver and its iteration scheme untouched. Additionally, it facilitates integration of the state and adjoint solver in an almost arbitrary state-of-the-art optimization approach. This means for example that globalization approaches can be directly applied. Furthermore, inequality constraints can be treated with an appropriate constrained optimization approach, such as the one described in chapter 4. For all these reasons, nested optimization methods are widely used for industrial applications [109]. However, the nested approach might not lead to the fastest convergence of the optimization problem. Indeed, demanding full convergence of the state and adjoint equations in the first optimization stage, far from the optimum, is redundant and will only slow down the optimization process. An alternative approach for PDE-constrained optimization was first introduced by Ta'asan, who proposed to treat the design equation rather like a boundary condition for the forward and adjoint partial differential equations [136].

This attractive alternative to the nested optimization approach was the first so-called *one-shot optimization* approach. In general, these methods aim at a simultaneous iterative solution of state, adjoint, and design equations. Specific to one-shot methods is that they again leave the iterative structure of the state

and adjoint solver untouched, and use them to construct an appropriate iterative solution procedure for the KKT conditions.

The promise of increased optimization efficiency motivates a further investigation of one-shot optimization methodologies. Many challenges have to be met though before one-shot optimization can be applied to magnetic divertor design and compete with the nested approach in terms of robust convergence and treatment of design constraints. First, in section 7.1, an overview of relevant existing one-shot approaches will be given to assist the reader. The remainder of the chapter addresses the challenges of one-shot optimization.

In section 7.2, the potential of one-shot will be tested by using a straightforward one-shot approach to an unconstrained magnetic divertor design problem. Although the unconstrained setting is somewhat artificial, this first study will give an indication of the one shot approach's challenges and gains. One of the conclusions is that the increased number of grid generations slows down the one-shot optimization process. Another conclusion is that robust convergence is not achieved due to discontinuous changes of the gradient values in design space. In section 7.3, the one-shot procedure will then be further accelerated. To this end, a grid deformation method for field aligned grids is developed.

In an attempt to robustify the one-shot procedure, section 7.4.2 discusses a novel strategy to achieve global convergence based on an augmented Lagrangian penalty function. The strategy is based on the doubly augmented Lagrangian function for one-shot optimization of Hamdi and Griewank [65, 66], but adapted to reduce computational costs and implementation efforts. In addition, a one-shot version of the BFGS technique is consistently derived, including a practical method to bound the Hessian. After motivating and explaining the novel methodology, section 7.5 discusses a number of adaptations to ensure the compatibility between the in-parts continuous adjoint sensitivity calculation, introduced in chapter 5, and the novel one-shot method. Results of this algorithm are then presented for an unconstrained divertor design problem. Although the approach is developed for magnetic divertor design, the novel one-shot method presented in this chapter is more generally applicable. By combining the global convergence of Griewank's method with reduced computational costs and implementation effort, the method offers an attractive alternative to other one-shot methods.

Finally, inequality constraints should be considered. In aerodynamics applications, one-shot methods have been adjusted to handle a single state constraint by either assuming the constraint to be active [55] or by using a penalty multiplier method [106]. However, in the present optimization problem with multiple design constraints, these approaches are not suitable. In section 7.6, an outlook is therefore given on a strategy to incorporate the design constraints in the one-shot approach. This constrained optimization strategy, however, remains to be verified in future research.[†]

7.1 Literature review on one-shot approaches

In section 2.2, one-shot methods were introduced as approximate rSQP methods that directly solve for the KKT conditions of the PDE-constrained optimization problem. Of course, one-shot methods are more than a direct Newton solution of the KKT system. Indeed, a vast amount of literature exists on the stable iterative solution of partial differential equations. Often, a stable solution of stationary PDEs is only found using pseudo-timestepping methods or strong relaxation. Considering again the generic PDE-constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{\varphi},\boldsymbol{q}} \quad & \mathcal{J}\left(\boldsymbol{\varphi},\boldsymbol{q}\right) \\ s.t. \quad & \boldsymbol{c}(\boldsymbol{\varphi},\boldsymbol{q}) = 0, \end{aligned} \tag{7.1}$$

the straightforward iterative solution of the approximate rSQP system

$$\begin{bmatrix} 0 & 0 & A^* \\ 0 & B & \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^* \\ A & \partial_{\boldsymbol{\varphi}}\boldsymbol{c} & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{q} \\ \Delta\boldsymbol{\varphi} \\ \Delta\boldsymbol{q}^* \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{q}}\mathcal{L} \\ -\nabla_{\boldsymbol{\varphi}}\mathcal{L} \\ -\boldsymbol{c} \end{bmatrix} \tag{7.2}$$

from equation (2.34) might therefore simply diverge. Only if the A, B, and A^* matrices include sufficient relaxation, stable convergence of the one-shot method can be achieved. Therefore, the matrices A and A^* preferably include the relaxation of the original state solver.

[†]Parts of this chapter have been submitted for publication in "Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N.R. & Reiter, D. (2017). A practical globalization of one-shot optimization for optimal design of tokamak divertors. *Journal of Computational Physics*, **328**, 399–412".

In fact, one-shot methods always aim at integrally reusing the original solver in an attempt to inherit its stable convergence. The first introduction of one-shot methods by Ta'asan as a novel boundary condition to the state and adjoint solver illustrates this best [136]. The method solves the equations

$$\begin{aligned} \frac{\partial \boldsymbol{q}}{\partial t} - \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) &= 0, \\ \frac{\partial \boldsymbol{q}^*}{\partial t} - \nabla_{\boldsymbol{q}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*) &= 0, \\ \nabla_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*) &= 0, \end{aligned} \tag{7.3}$$

using the usual pseudo-time stepping iterations for state and adjoint equations followed by a solution of the design equation. Hazra et al. [68, 69] later advocated a simultaneous pseudo-time stepping framework, in which also the design equation was relaxed using pseudo-time stepping. The equations

$$\begin{aligned} \frac{\partial \boldsymbol{q}}{\partial t} - \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) &= 0, \\ \frac{\partial \boldsymbol{q}^*}{\partial t} - \nabla_{\boldsymbol{q}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*) &= 0, \\ \frac{\partial \boldsymbol{\varphi}}{\partial t} - \nabla_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*) &= 0. \end{aligned} \tag{7.4}$$

are then said to be preconditioned with an approximate rSQP matrix in the form of equation (7.2) and iteratively solved.

The approach has been proven efficient for a number of aerodynamics design problems [69, 70, 127]. Although this is an interesting approach, it should be noted that whereas the adjoint equations inherit the time scale of the state equations, the design equations do not. One might therefore wonder what an appropriate time scale for the design equation is. In the end, one thus still ends up looking for an appropriate design relaxation. Moreover, it might be questioned under which conditions this approach leads to convergence. A first step in the direction of a convergence proof was taken in Ref. [79], where convergence of linearly constrained quadratic optimization problems has been addressed.

In a more general context of fixed-point solvers, Griewank and Faure [64] promoted simultaneous solution of state and adjoint equation using so-called *piggyback* iterations. The name 'piggyback' stems here from the fact that the state is continuously varying in the adjoint equations. The adjoint solvers is thus

heading for a moving target and depends on state convergence to converge. These piggyback iterations are especially useful when AD is applied, since functions and derivatives are then evaluated simultaneously.

A subsequent publication extended the approach to one-shot optimization and necessary conditions on the design preconditioner B were derived [62]. Eventually, Hamdi and Griewank [65] obtained a local convergence proof using a design preconditioner B based on a *doubly augmented Lagrangian* merit function. The convergence proof is then induced by proving descent of the one-shot method on this penalty function that augments the Lagrangian with state and adjoint feasibility penalties. Where the approach was illustrated in Ref. [65] for an academic problem, Özkaya and Gauger first applied this new preconditioner for aerodynamic shape optimization [104, 105]. The approach was afterwards extended to global convergence by introducing line search algorithms based on this doubly augmented Lagrangian merit function [66]. Further extensions comprise the inclusion of additional design constraints [147] and an extension of the convergence theory to functional spaces [81, 82], allowing for adaptive grid refinement.

Bosse et al. finally examined so-called *multi-step* one-shot methods in the fixed-point framework [23]. In these multi-step methods a design step is no longer followed by a single state and adjoint step. Instead, multiple state and adjoint iterations are performed after each design step. A lower bound for the number of state and adjoint iterations per design step was then derived for local convergence. An overview of some of these recent advances and some applications can be found in Ref. [22].

In sections 7.2 and 7.3, a simple one-shot method based on the approach of Hazra et al. is used for first tests. Afterwards, in section 7.4, the globalized one-shot method of Hamdi and Griewank [65, 66] will be further discussed and will serve as a basis for the introduction of a novel one-shot procedure. Throughout the chapter, all optimization approaches are introduced in the generic optimization framework (7.1) for easy referencing. It should be noted that the interpretation in the in-parts adjoint framework of chapter 5 is easily done by replacing the state $\boldsymbol{q}$ by the plasma state $\boldsymbol{q}_{\mathrm{pe}}$. Similarly, model equations $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) = 0$, cost function $\mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q})$, and Lagrangian $\mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*)$ should be replaced by $\widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) = 0$, $\widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})$, and $\widetilde{\mathcal{L}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}})$, respectively.

7.2 A first attempt to one-shot magnetic divertor design

7.2.1 One-shot approach

In a first step, the potential of one-shot algorithms for magnetic divertor design is tested with an approach similar to that described by Hazra et al. [68, 69]. The one-shot algorithm performs the following steps in each iteration.

Algorithm 7.1. *A simple unconstrained one-shot optimization algorithm*

1. *Make a pseudotime-step Δt in the state equations and update $\boldsymbol{q}$ by performing one step on the state equations.*

2. *Make a pseudotime-step Δt in the adjoint equations and update $\boldsymbol{q}^*$ by performing one step on the state equations.*

3. *Using $\boldsymbol{s_k} = \boldsymbol{\varphi}_k - \boldsymbol{\varphi}_{k-1}$ and $\boldsymbol{y_k} = \nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_k) - \nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_{k-1})$ from the last two iterations, calculate the design preconditioner B_k from*

$$B_k = \bar{\beta}\frac{\boldsymbol{s_k}^\top \boldsymbol{y_k}}{\boldsymbol{s_k}^\top \boldsymbol{s_k}}\delta_{ij}, \qquad \text{with} \qquad \beta_{min} < \frac{\bar{\beta}\boldsymbol{s_k}^\top \boldsymbol{y_k}}{\boldsymbol{s_k}^\top \boldsymbol{s_k}} < \beta_{max}, \tag{7.5}$$

   and with $\bar{\beta}$ a constant.

4. *Make a design step by calculating $\boldsymbol{\varphi}_{k+1}$ from $\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - B_k^{-1}\nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_k)$.*

In this algorithm, the exact reduced gradient $\nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_k)$ is obviously never available during one-shot optimization. The estimation $\nabla\hat{\mathfrak{J}}(\boldsymbol{\varphi}_k) \approx \nabla_\varphi \mathcal{L}(\boldsymbol{\varphi}_k, \boldsymbol{q}_k, \boldsymbol{q}_k^*)$ is therefore used. This estimate improves as state and adjoint equations converge. It can also be seen that the estimation of B from Eq. (7.5) will merely lead to a scaled steepest descent step. Therefore, it will most probably not lead to the computationally most efficient algorithm. Nevertheless, it allows the straightforward introduction of Hessian bounds that contribute to the robustness of the algorithm.

In this first test, conservative bounds are chosen on the one-shot approach. In the philosophy of simultaneous pseudotime-stepping, one might choose the Hessian bounds β_{min} and β_{max} such that state and optimization convergence go

at about the same pace. The order of magnitude for this Hessian relaxation (for the normalized control variables in the code) can then be obtained by dividing the typical number of iterations needed for state convergence by the number of optimization iterations typically needed for nested optimization. One then arrives at typical values of about 10^4. The bounds are therefore chosen as $\beta_{min} = 2000$ and $\beta_{max} = 20000$. $\bar{\beta}$ is set to 1 for now, but can be increased to have additional relaxation in the design steps. The one-shot procedure is started from a converged state and costate for the initial configuration.

7.2.2 Set-up of an unconstrained WEST test case

The test case for this first one-shot study is based on the WEST case of the previous chapter. The perturbation model is used for magnetic field calculations, to reduce computational efforts and to simplify the setting for this first study as much as possible. Since the introduction of design constraints clearly poses additional challenges for the one-shot approach, an unconstrained framework is favoured for this study. The design constraints $\boldsymbol{h}(\boldsymbol{\varphi}) \leqslant 0$ are therefore removed at this stage. As discussed, however, they are crucial since they avoid the occurrence of configurations that cause code crashes. Therefore, the cost functional is modified with a penalty of augmented Lagrangian type. That is, the original cost functional from (4.4) is replaced by

$$\mathcal{J}^A(\boldsymbol{\varphi}, \boldsymbol{q}) = \mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}) + \mu \boldsymbol{h}_{\mathbb{A}} + \eta \|\boldsymbol{h}_{\mathbb{A}}\|^2, \tag{7.6}$$

with $\boldsymbol{h}_{\mathbb{A}}$ the active contraints, and μ and η weighting parameters of the penalty terms. If μ approaches the Lagrangian multipliers of the constrained optimization problem and if η is chosen sufficiently big, this is an exact penalty function and the optimum of the original constrained optimization problem is a stable optimum of the new penalty function (7.6) [103]. From a nested optimization study, it is found that the maximal current limit on coil 14 is the only active constraint at the optimum. Its corresponding Lagrange multiplier $\mu = 0.9$ can thus be adopted. The quadratic penalty parameter is set to $\eta = 10$. Of course, further developments towards a constrained one-shot approach will later be necessary after the application to this unconstrained problem is proven successful. Since

this unconstrained setting is used for the one-shot tests throughout this chapter, the augmented objective $\mathcal{J}^A$ will simply be referred to as the objective $\mathcal{J}$.

7.2.3 Discussion of first results

Starting from a converged state and costate, 5000 iterations of the one-shot method are performed for a first convergence check. Notice that this is a low number of iterations since full state convergence would require at least around 20000 iterations. Yet, it is sufficient to see whether the one-shot approach tends to converge and to highlight some features of the one-shot method. The main reason for applying this approach is the high computational cost as discussed below.

In figure 7.1a, the feasibility of the different equations is analysed as a function of the one-shot iteration count. State and adjoint feasibility are represented by their scaled residuals. The feasibility of the design equation is represented by the norm of the design gradient $\|\nabla_{\boldsymbol{\varphi}}\mathcal{L}\|_2$. When analyzing the results, disturbing jumps in residuals immediately draw the attention. Nevertheless, the cost functional in figure 7.1b initially shows a good decrease and stagnates afterwards, dominated by the gradient noise. However, the one-shot method is not able to converge smoothly here, since the occurrence of the large gradient jumps leads to sudden increases of the design steps and associated rises in state and adjoint residuals.

Since these jumps thus impede one-shot convergence, a logical next step is to trace their source and see whether they can be avoided. In figure 7.2, the design gradient is shown for a nested optimization of the same unconstrained problem. A similar jump in the design gradient is observed here at the third iteration, although this does not prevent further optimality convergence. A quick scan of objective and its directional derivative along the design step in the third iteration (at constant state and adjoint) reveals that discontinuous gradient changes lie at the basis of the problem. This can be clearly seen from figure 7.3, where objective and directional derivative are depicted as a function of relative step size.

Studying this discontinuity in the design gradient $\nabla_{\boldsymbol{\varphi}}\mathcal{L}$ further in depth, the source is pinpointed in the grid generator part of the code. As discussed in

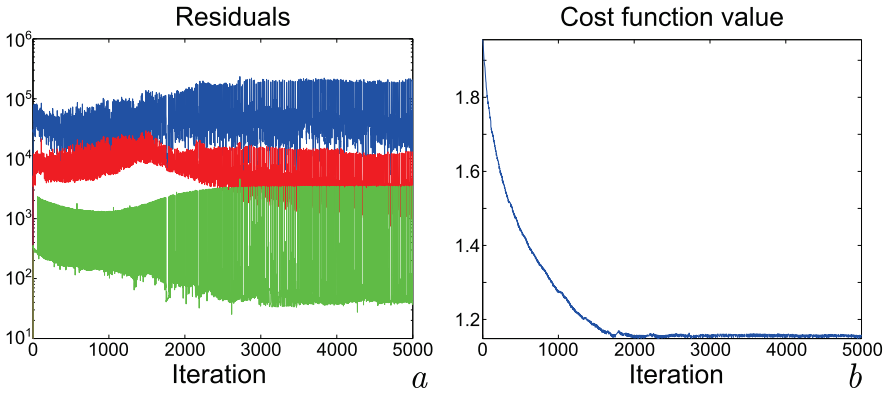


Figure 7.1: Convergence of the first one-shot optimization attempt. The left figure shows (from top to bottom) the state (blue) and adjoint (red) scaled residuals, and the scaled design feasibility $\|\nabla_{\varphi}\mathcal{L}\|_2$ as a function of one-shot iterations. Discontinuities in $\nabla_{\varphi}\mathcal{L}$ clearly impede proper one-shot convergence, though the cost function (right) initially decreases normally as a function of the one-shot iteration count.

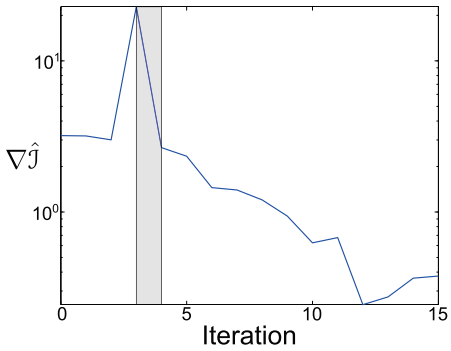


Figure 7.2: The reduced gradient $\nabla\hat{\mathcal{J}}$ is shown during nested optimization convergence of the same unconstrained test case. Notice the gradient jump around iteration 3.

chapter 6, the outer grid boundary in the SOL is either chosen at a magnetic field line that includes the second X-point, or at the first field line that touches a

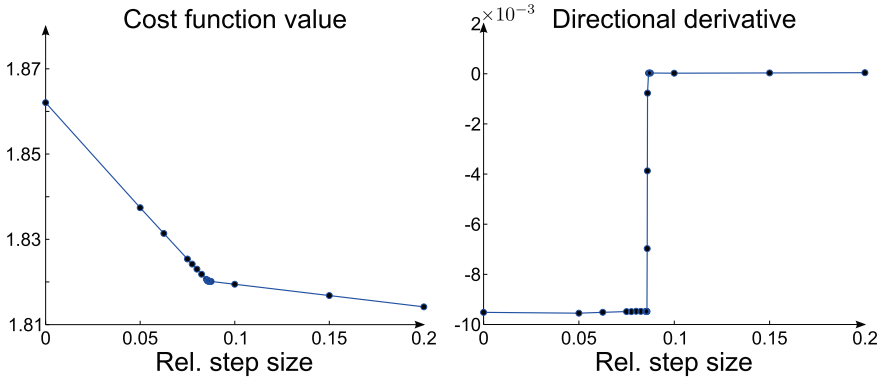


Figure 7.3: Analysis of the cost function and directional derivative between iteration 3 and 4 of the nested optimization study (indicated grey in figure 7.2) shows a discontinuous derivative. The left figure shows the cost function value as a function of the relative step size and evaluated at constant plasma state. The right figure shows its discontinuous derivative (evaluated at constant plasma state and costate). The dots represent the values at which these quantities where evaluated.

vessel boundary different from the target plates. Coincidence wants that in this case, these two field lines appear to coincide during optimization (see figure 7.4). This results in regular switching between the two possible grid boundaries. Since the location of the 2$^{\text{nd}}$ X-point is strongly influenced by the divertor coil currents, the switching gives rise to an associated strong change in design sensitivity.

Finally, the one-shot algorithm can be compared to the nested solution of the same unconstrained problem. After 16 optimization iterations, the nested optimization algorithm reaches the same cost function value ($\mathcal{J} = 1.16$) as the one-shot algorithm. Notice that, in principle, the cost function value can still change if state and adjoint equations are converged at the final design. Nevertheless, this indicates that the one-shot method tends to find the right direction. The evolution of the design variables is shown in figure 7.5. A similar trend is found here as well for nested and one-shot optimization. Given the similarity, it can be informative to compare the computational cost spent so far in the two approaches. In table 7.1, it can be seen that while the nested approach needs 670000 iterations on state and adjoint equations, only 10000 are needed for one-shot. Of

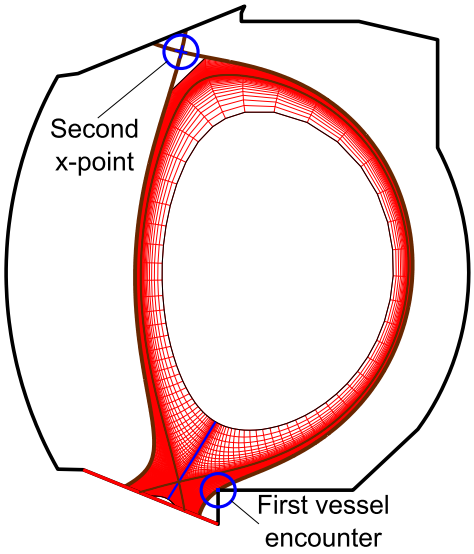


Figure 7.4: An illustration of the grids that are found near gradient discontinuities. Notice that the two means to choose the outer SOL grid boundary, the second X-point location or the first vessel encounter of a magnetic field line in between two target plates, coincide. This causes the discontinuities in the gradient observed in figure 7.1.

course, evaluations of $\hat{\mathcal{J}}$ and $\nabla\hat{\mathcal{J}}$ in line search contribute to the total number of function evaluations during nested optimization. It therefore seems the potential gain in computational cost of using one-shot optimization is enormous (approximately a factor 70). These differences are, however, much less pronounced in wall-clock time. The wall-clock time indicated in table 7.1 is given for the 20-core workstation with 2 Intel® Xeon® processors of type E5-2670 v2, on which the routines are run. It should be noted that the values are purely indicative since it is subject to coincidental parameters like the amount of jobs being processed. Although a speed-up of about a factor 3 in wall-clock time is indicated in the table for one-shot optimization, it only obtains this speed-up if the finite difference parts of the gradient calculations are parallelized. The obvious reason that the reduction in iterations is not reflected as pronounced in wall-clock time

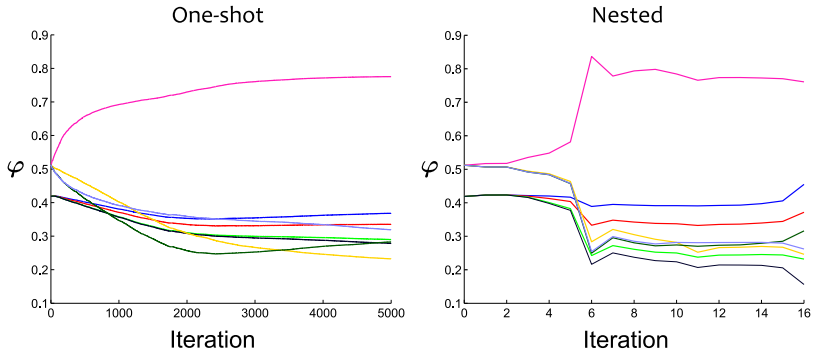


Figure 7.5: Preliminary comparison of the design variable history of the one-shot optimization approach with that of the nested approach. Remark that the comparison is only indicative since design variables are still evolving in both the one-shot and the nested optimization approach. In comparison, optimization iterations in the nested approach demand 40000 system solves of state and costate for each design step, while one-shot only demands two.

(and not at all in CPU-time) is the significant increase in in-parts adjoint design gradient evaluations. While the computational cost of these gradient calculations was negligible in the nested optimization approach, they dominate in the one-shot approach. In comparison, state and adjoint equations jointly take about 3 seconds per iteration, while evaluation of $\nabla_{\varphi}\mathcal{L}$ takes slightly more than 30 seconds for each design variable when approximated by central difference calculations. A closer examination reveals that grid generation accounts for the majority of the gradient evaluation cost.

In conclusion, the one-shot method shows bad convergence behaviour because of the non-smooth choice of grid boundary. It should be noted that this boundary is artificially chosen in absence of state-of-the-art plasma edge techniques calculating up to the vessel wall [5, 27] and should therefore not appear if optimization is combined with such a code. Given the huge potential gain in computational cost of one-shot methods, one-shot divertor optimization will be studied further in the next section in absence of this gradient discontinuity near the optimum. However, the high computational cost from evaluating the in-parts adjoint gra-

	Nested	One-shot
# state/adjoint iterations	670000	10000
# gradient evaluations	39	5000
wall-clock time (hours)	320	109

Table 7.1: Comparison of nested and one-shot optimization efficiency. From top to bottom, total number of state and adjoint iterations, design gradient evaluations $\nabla_{\boldsymbol{\varphi}}\mathcal{L}$, and wall-clock time are listed for the nested and one-shot optimization studies in figure 7.5. The wall-clock time is given for the 20-core workstation with 2 Intel® Xeon® processors of type E5-2670 v2, on which the routines are run and is subject to coincidental parameters like the amount of jobs being processed. Gradient calculations are parallelized for one-shot optimization.

dient in each one-shot iteration is tackled first. To this end, a more efficient alternative to grid generation will be developed.

7.3 Acceleration of the one-shot procedure

It is clear that in one-shot optimization, the increased number of sensitivity calculations urges for faster sensitivity calculations. Indeed, the argument from chapter 5 that the influence of magnetic field calculation and mesh generation on the total computational cost is relatively low does no longer hold. Although the in-parts adjoint procedure is kept for now, a reduced CPU-time associated to computing grid derivatives and design updates is aimed for by evaluating $\boldsymbol{q}_{\mathrm{gg}}(\psi)$ using a grid deformation algorithm instead of a full mesh generation step. Grid deformation methods have shown their virtue both for moving boundary problems and for optimal design[49]. Since no such method exists for field aligned grids, a grid deformation methodology for structured curvilinear field-aligned grids is elaborated in this section. More specifically, a grid deformation method is developed based on the spring deformation analogy [8, 10].

7.3.1 Grid deformation method

The objective of the grid deformation method is twofold. On the one hand, the grid deformation method needs to realign an original grid with node locations $\boldsymbol{x_G}$ to the new magnetic flux function ψ. On the other hand, it needs to ensure that the grid is sufficiently smooth and orthogonal. In figure 7.6, the concept of the spring deformation method is illustrated. The initial grid is deformed to match the new flux function, while the equilibrium in the springs is retrieved. Remark that because of the nonlinear flux function ψ, an iterative approach is necessary to tackle the spring deformation problem. For the radial movement of the grid nodes, the nonlinearity of ψ necessitates successive linearizations to find node locations that lie on the new flux surfaces. If the nodes lie on the desired flux surfaces, the problem reduces to moving the nodes along the flux surfaces to retrieve the spring equilibrium. For this poloidal movement, the curved geometry of the flux surfaces is an additional source of nonlinearity that needs linearization. The iterative grid deformation proposed below linearizes both radial and poloidal movements and moves the nodes in both directions until both criteria are met.

Let $\boldsymbol{x} = \boldsymbol{R}\,\mathbf{e}_\mathrm{R} + \boldsymbol{Z}\,\mathbf{e}_\mathrm{Z}$ be the Cartesian coordinates of a set of grid nodes that satisfies the spring equilibrium on the new flux surfaces. The grid deformation solver then seeks the movement $\boldsymbol{dx} = \boldsymbol{dR}\,\mathbf{e}_\mathrm{R} + \boldsymbol{dZ}\,\mathbf{e}_\mathrm{Z}$, which governs the transformation between an initial grid $\boldsymbol{x_0}$ and the deformed grid $\boldsymbol{x} = \boldsymbol{x_0} + \boldsymbol{dx}$. A robust choice of the initial node locations $\boldsymbol{x_0}$ will be further discussed at the end of this section.

Expressing the movement $\boldsymbol{dx}$ of the grid nodes in the curvilinear coordinate system $(\mathbf{e}_\theta, \mathbf{e}_r)$ aligned with the magnetic field will significantly simplify the expressions of the grid deformation solver and will lead to a block diagonal deformation matrix as will be shown further. Considering a linearized step $\boldsymbol{\Delta x_k}$, the grid deformation $\boldsymbol{dx}$ is then approximately found by $\boldsymbol{dx_n}$ after n iterations that solve for the linearized steps $\boldsymbol{\Delta u_k}$ and $\boldsymbol{\Delta v_k}$, in the poloidal and radial directions, respectively:

$$\boldsymbol{dx} \approx \boldsymbol{dx_n} = \sum_{k=1}^{n} \boldsymbol{\Delta x_k} = \sum_{k=1}^{n} \boldsymbol{\Delta u_k}\ \mathbf{e}_\theta(\boldsymbol{x_k}) + \boldsymbol{\Delta v_k}\ \mathbf{e}_r(\boldsymbol{x_k}). \tag{7.7}$$

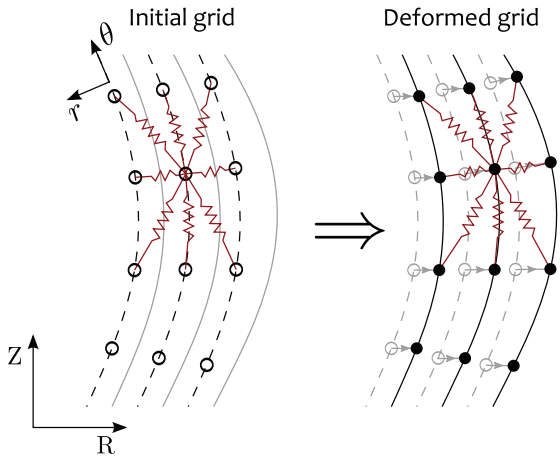


Figure 7.6: The concept of the novel grid deformation method for flux-aligned curvilinear coordinate systems. Starting from an initial grid with grid nodes (empty dots) lying on an initial set of flux surfaces (dashed lines), the grid deformation method adapts the node positions to lie on the flux surfaces of a new poloidal flux function ψ (solid lines). Simultaneously, the final node positions (solid dots) retrieve force balance along the magnetic flux surfaces in a system of springs that connects neighbouring nodes. The springs connected to one particular node are illustrated in red.

Here, $\mathbf{e}_\theta(\boldsymbol{x}_\mathrm{k})$ and $\mathbf{e}_r(\boldsymbol{x}_\mathrm{k})$ are the Cartesian unit vector representations of the $\mathbf{e}_\theta$ and $\mathbf{e}_r$ directions at locations $\boldsymbol{x}_\mathrm{k}$.

Let us now consider the linearized equations after l iterations of this iterative solver. The first condition demands alignment of the grid edges with the magnetic field. This can be solved for with Newton's method (at least for limited magnetic field changes) once one knows on which flux lines the grid nodes must lie. The desired flux values for each node of the grid are combined in a vector $\boldsymbol{\psi_d}$. A linearized deformation step $\boldsymbol{\Delta v_l}$ can then be solved for as

$$[\nabla_r \psi]_l \ \boldsymbol{\Delta v_l} = \boldsymbol{\psi_d} - \boldsymbol{\psi_l}, \tag{7.8}$$

with $\boldsymbol{\psi_l}$ and $[\nabla_r \psi]_l$ the values of flux function ψ and its radial gradient $\nabla_r \psi$ at the nodes $\boldsymbol{x_l}$. Remark that the poloidal movement $\boldsymbol{\Delta u_l}$ does not appear, since

$\nabla_\theta \psi \equiv 0$.

Finding the poloidal movement is somewhat more complex. The grid should be stretched to match the same domain boundaries (the target surfaces) again, while it retains the refinements towards crucial zones (targets, separatrix,...). For this purpose, the spring deformation method is used [8, 10]. Each node i is connected to its N neighbouring nodes $j = 1 : N$ ($N = 8$ in this case) using fictitious springs with linear deformation constant

$$\mathcal{K}_{i,j} = \frac{1}{L_{ij,\text{init}}}, \tag{7.9}$$

and with $L_{ij,\text{init}}$ the initial length of the springs. Since the radial movement (in direction $\mathbf{e}_r$) is constrained by the demand that the grid cells lie on the flux lines $\boldsymbol{\psi_d}$, the spring equilibrium is then solved as if the grid nodes can only slide along bars of constant ψ. Static equilibrium is then found by demanding that the poloidal force $F_\theta = F_R\ (\mathbf{e}_\mathrm{R} \cdot \mathbf{e}_\theta) + F_Z\ (\mathbf{e}_\mathrm{Z} \cdot \mathbf{e}_\theta) = 0$ at all nodes. Denote the component of the linearized step $\boldsymbol{\Delta x_l}$ at node i as $\boldsymbol{\Delta \zeta_i}$. Additionally, consider the spatial decomposition of this component using non-boldfaced symbols as

$$\boldsymbol{\Delta \zeta_i} = \Delta R_i\, \mathbf{e}_\mathrm{R} + \Delta Z_i\, \mathbf{e}_\mathrm{Z} = \Delta u_i\, \mathbf{e}_{\theta,i} + \Delta v_i\, \mathbf{e}_{\theta,i}. \tag{7.10}$$

The iteration number l is left out from these expressions for ease of notation. Similarly, consider the component $\boldsymbol{d\zeta_i}$ at node i of the total deformation $\boldsymbol{dx_l}$ from all previous iterations $1, \ldots, l$ according to (7.7) as

$$\boldsymbol{d\zeta_i} = dR_i\, \mathbf{e}_\mathrm{R} + dZ_i\, \mathbf{e}_\mathrm{Z}. \tag{7.11}$$

Given Hooke's law for the spring forces, the requirement for static equilibrium at node i is then given by

$$F_{\theta,i} = \sum_{j=1}^{N} \mathcal{K}_{i,j}\, (dR_i - dR_j)\ (\mathbf{e}_\mathrm{R} \cdot \mathbf{e}_{\theta,i}) + \sum_{j=1}^{N} \mathcal{K}_{i,j}\, (dZ_i - dZ_j)\ (\mathbf{e}_\mathrm{Z} \cdot \mathbf{e}_{\theta,i}) = 0, \tag{7.12}$$

with N the number of neighbouring nodes. Linearizing this equation in the $(\mathbf{e}_\theta, \mathbf{e}_r)$ coordinate system leads to

$$\frac{\partial F_{\theta,i}}{\partial \theta} \Delta u_i = \sum_{j=1}^{N} \mathcal{K}_{i,j} \left(\Delta u_i - \Delta u_j\, (\mathbf{e}_{\theta,j} \cdot \mathbf{e}_{\theta,i}) - \Delta v_j\, (\mathbf{e}_{r,j} \cdot \mathbf{e}_{r,i}) \right), \tag{7.13}$$

where the iteration number k is left out for ease of notation. Rewriting the spring equilibrium at node i, $F_{\theta,i} + \dfrac{\partial F_{\theta,i}}{\partial u}\Delta u_i = 0$, to isolate all sources of the equation on the right hand side finally gives

$$\sum_{j=1}^{N} \mathcal{K}_{i,j} \left(\Delta u_i - \Delta u_j \left(\mathbf{e}_{\theta,j} \cdot \mathbf{e}_{\theta,i}\right)\right) = \sum_{j=1}^{N} \mathcal{K}_{i,j}\, \Delta v_j \left(\mathbf{e}_{r,j} \cdot \mathbf{e}_{\theta,i}\right) - \tag{7.14}$$
$$\sum_{j=1}^{N} \mathcal{K}_{i,j} \left(dR_i - dR_j\right)\ \left(\mathbf{e}_{\mathrm{R}} \cdot \mathbf{e}_{\theta,i}\right) - \sum_{j=1}^{N} \mathcal{K}_{i,j} \left(dZ_i - dZ_j\right)\ \left(\mathbf{e}_{\mathrm{Z}} \cdot \mathbf{e}_{\theta,i}\right).$$

After solving equation (7.8) for $\boldsymbol{\Delta v_l}$, this last equation can be solved for the components of $\boldsymbol{\Delta u_l}$.

The boundary conditions of this equation are given by fixing the position of the target nodes. Starting from a position $\boldsymbol{X_1}$ of the node, the movement Δu_b of a boundary node is given by solving for the intersection with a section of the target polygon $[\boldsymbol{X_t}, \boldsymbol{X_t} + \Delta \boldsymbol{X_t}]$ (see figure 7.7). The intersection can be found from the system of equations

$$\begin{cases} \boldsymbol{X_1} + \Delta u_b\, \mathbf{e}_{\theta,1} + \Delta v\, \mathbf{e}_{r,1} = \boldsymbol{X_0} \\ \boldsymbol{X_t} + t\, \Delta \boldsymbol{X_t} = \boldsymbol{X_0} \end{cases} \tag{7.15}$$

that solves for the intersection point $\boldsymbol{X_0}$. Eliminating t and X_0 and substituting $\boldsymbol{X_2} = \boldsymbol{X_1} + \Delta v\, \mathbf{e}_{r,1}$ yields

$$\left(\frac{R_\theta}{\Delta R_t} - \frac{Z_\theta}{\Delta Z_t}\right) \Delta u_b = \frac{R_t - R_2}{\Delta R_t} - \frac{Z_t - Z_2}{\Delta Z_t}, \tag{7.16}$$

where R_θ and Z_θ represent the R and Z components of the unit vector $\mathbf{e}_\theta$, respectively. Similarly, the Cartesian decompositions of the vectors $\boldsymbol{X_t}$, $\Delta \boldsymbol{X_t}$, and $\boldsymbol{X_2}$ are given by $\boldsymbol{X_t} = R_t \mathbf{e}_{\mathrm{R}} + Z_t \mathbf{e}_{\mathrm{Z}}$, $\Delta \boldsymbol{X_t} = \Delta R_t \mathbf{e}_{\mathrm{R}} + \Delta Z_t \mathbf{e}_{\mathrm{Z}}$, and $\boldsymbol{X_2} = R_2 \mathbf{e}_{\mathrm{R}} + Z_2 \mathbf{e}_{\mathrm{Z}}$, respectively. In conclusion, since $\boldsymbol{\Delta v_l}$ can be solved for independently by (7.8), a sequential solver can be used that iteratively solves for the radial movement $\boldsymbol{\Delta v_l}$ and the poloidal movement $\boldsymbol{\Delta u_l}$ from the spring equilibrium equations (7.14) with boundary conditions given by (7.16).

It should be noted that the choice of the initial grid node locations $\boldsymbol{x_0}$ is not trivial for this grid deformation method. Whereas in typical grid deformation solvers $\boldsymbol{x_0}$ are the points of the original node locations, a similar approach would

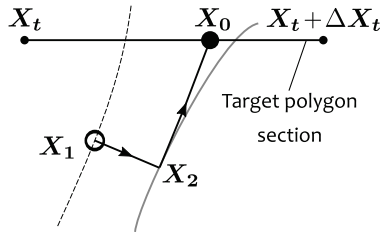


Figure 7.7: Illustration of the boundary condition for the poloidal node movement $\boldsymbol{\Delta u}$. The nodes are shifted to lie on the target surface.

lead to unwanted results here. That is, if the magnetic field is altered, the so-called X-point location is changed. Recall, however, that the X-point is a saddle point of the magnetic flux function ψ. Therefore, for each value ψ_d close to the flux function value of the X-point ψ_X, two solutions can be found in the radial direction. Namely, one at each side of this saddle point. If for the novel magnetic flux ψ, some of the nodes at initial locations $\boldsymbol{x_0}$ are not at the correct side of the X-point anymore, (7.8) will converge for these nodes to a location at the wrong side of the X-point. Therefore, one cannot simply use the locations $\boldsymbol{x_G}$ of the initial grid as initial node locations. To circumvent the problem, the initial node locations are chosen as

$$\boldsymbol{x_0} = \boldsymbol{x_G} + (r_X - r_{X,G})\,\mathbf{e}_\mathrm{R} + (z_X - z_{X,G})\,\mathbf{e}_\mathrm{Z}, \tag{7.17}$$

with $\boldsymbol{X_X} = r_X\,\mathbf{e}_\mathrm{R} + z_X\,\mathbf{e}_\mathrm{Z}$ the new location of the X-point and $\boldsymbol{X_{X,G}} = r_{X,G}\,\mathbf{e}_\mathrm{R} + z_{X,G}\,\mathbf{e}_\mathrm{Z}$ the location of the X-point on which the original grid was constructed. Remark that when multiple X-points are present, one might need to extend this approach to shift the initial locations only in a narrow region around the X-points.

The grid deformation procedure gives significant speed-up with respect to full mesh generation. Using grid deformation, the total computational time of the grid construction is reduced by approximately a factor of ten. Nevertheless, larger deformations might lead to qualitatively inferior grids. For the WEST configuration, presented in the next section, such grid irregularities are mainly situated at the crossing of vessel structure and private flux grid boundary. In literature, a local stiffening of the springs is proposed to circumvent these irregularities near

boundaries [10]. In our algorithm, a stiffening of the poloidal springs (those connecting 2 points on the same flux line) on the first couple of flux lines near the private flux boundary has proven the most effective way to resolve the problem. In the optimization procedure, qualitatively inferior grids are easily avoided by creating a new grid after a fixed number of (one-shot) optimization iterations or when the design change since the last mesh generation event exceeds a limit, that is $\|\boldsymbol{\varphi} - \boldsymbol{\varphi}_G\| > \zeta$, with $\boldsymbol{\varphi}_G$ the control variables associated to the last constructed grid with node locations $\boldsymbol{x_G}$ and ζ a fixed code parameter. Finally, the grid deformation method is a step further towards future extension of the in-parts adjoint approach to full adjoint sensitivity calculations including grid deformation (and eventually magnetic field calculation), similar to the procedure found in Ref. [102]. The further elaboration of this full adjoint approach, however, still requires a lot of derivational effort and is therefore left for future work. The grid deformation method will be used to accelerate design updates and (in-parts adjoint) sensitivity calculations in the further tests on one-shot optimization.

7.3.2 Adapted test case

It is clear that with the grid deformation method at hand, the computational efficiency of the one-shot method is much more competitive. A remaining problem is the presence of the gradient discontinuities. In an attempt to change the case as little as possible but avoiding the discrete switching of grid boundaries, the currents in coils 12 and 13 (see figure 6.2) are set to the optimum of the nested optimization study, while the currents in coils 10 and 11 are kept fixed to their initial values. This choice of currents aims at keeping the grid boundary as far as possible from the gradient discontinuity in the initial stage of the optimization, while the inequality constraints are still likely to be obeyed at the novel optimum using the same penalty parameters in the objective function $\mathcal{J}^A$.

Using the grid deformation method for design updates and within the in-parts adjoint sensitivity evaluations, the one-shot method from the previous section is now used to optimize the lower divertor coils (coils 14-17). This time, the one-shot algorithm consists of three stages. First, state and adjoint iterations are alternated to converge the initial state. This is the so-called *piggybacking*

technique described in section 7.1. Then, a one-shot optimization follows. When optimization stagnates, piggybacking is again used to converge state and adjoint equations. These stages are clearly indicated in the figures using roman numbers I, II, III.

7.3.3 Results and discussion

Convergence behaviour

In figure 7.8, the evolution of the (in-)feasibility of state, adjoint, and design equation is shown. Because of the changed test case, gradient discontinuities related to grid boundary switching are now only retrieved entirely at the end of the optimization cycle II (around iteration 21000). Smaller discontinuities do, however, still seem to appear in the course of one-shot optimization. This can be seen by analyzing the design feasibility $\|\nabla_{\boldsymbol{\varphi}}\mathcal{L}\|_2$ (green curve in figure 7.8). Looking at state and adjoint residuals, one notices that they remain very noisy. Partially, these residual jumps are due to the regular remeshing. The sudden grid change causes the residuals to increase instantaneously, but they decrease quite quickly again afterwards. Notice that the occurence of remeshing events is shown just above figure 7.8. The other sources of gradient discontinuities have not been traced. In a mesh generator, they are probably hard to entirely eliminate since linear interpolations, for instance, are omnipresent. Nevertheless, better convergence is already observed in comparison to the first trial. Moreover, compared to a nested algorithm that optimizes the same unconstrained test case without grid deformation, the match in both control variables and cost function value at the optimum is good. This can be seen in table 7.2, by comparing the first two columns.

Optimization efficiency

Given the very strict choice of relative step bounds β_{min} and β_{max}, the used optimization method could be interpreted as a glorified one-shot equivalent of the steepest descent method. Further acceleration of optimality convergence could therefore be obtained if these step bounds are “loosened” and eventually opting for a more refined Hessian estimation technique than equation (7.5). As such,

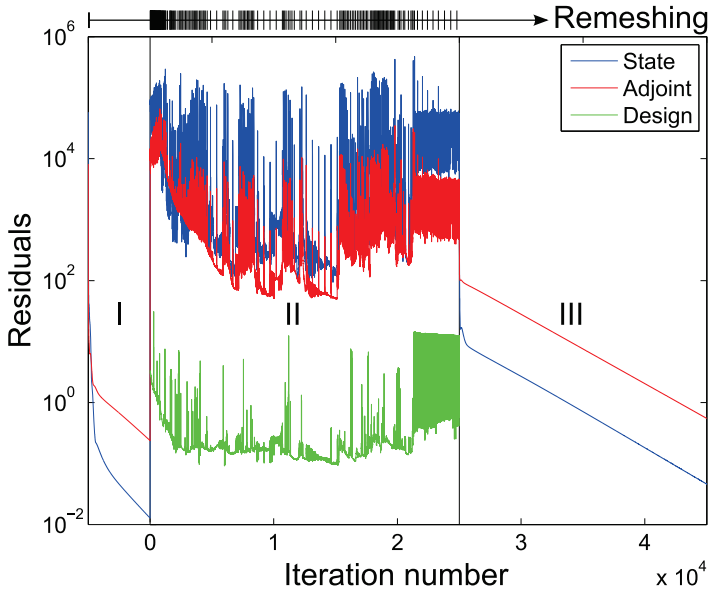


Figure 7.8: Convergence plots for an adjusted one-shot optimization attempt including the novel grid deformation method. The figure shows (from top to bottom) the state (blue) and adjoint (red) residuals, and the design feasibility $\|\nabla_{\varphi}\mathcal{L}\|_2$ (green) as a function of one-shot iterations. Before and after one-shot optimization, piggybacking iterations are used to simultaneously converge state and adjoint residuals. Remeshing events are indicated above the figure.

superlinear convergence of the optimization procedure might be found. The relaxation of these step bounds is therefore tested on the same case. In table 7.2, the parameters $\bar{\beta}$, β_{min}, and β_{max} are listed for these two tests, along with the final cost function value and design variables after optimization and state convergence.

The values β_{min} and β_{max} in the first additional test called 'one-shot 2' are chosen to relax both the minimal and maximal Hessian bounds in the algorithm. By decreasing β_{min} from $2 \cdot 10^3$ to $1 \cdot 10^2$ with respect to 'one-shot 1', the maximal design step increases by a factor of 20. To avoid that this increased step size would lead to divergence of the one-shot optimization, $\bar{\beta}$ is also increased by a factor of two to give some more design relaxation. Since increasing the upper Hessian

bound β_{max} would not directly be expected to trigger convergence problems at first sight, it is increased by a factor 250. In the 'one-shot 3' case, β_{min}, β_{max}, and $\bar{\beta}$ are all increased by a factor of ten with respect to 'one-shot 2'. This is equivalent to relaxing the design step size by an additional factor of ten. These two additional one-shot tests are run for the same amount of one-shot iterations (25000) as the 'conservative' one-shot test 'one-shot 1'.

	Nested	one-shot 1	one-shot 2	one-shot 3
$\bar{\beta}$	\	1	2	20
β_{min}	\	$2 \cdot 10^3$	$1 \cdot 10^2$	$1 \cdot 10^3$
β_{max}	\	$2 \cdot 10^4$	$5 \cdot 10^6$	$5 \cdot 10^7$
$\hat{\mathfrak{J}}(\bar{\boldsymbol{\varphi}})$	1.08	1.07	1.07	1.18
$\bar{\varphi}_1$	0.76	0.77	0.78	0.75
$\bar{\varphi}_2$	0.07	0.08	0.07	0.28
$\bar{\varphi}_3$	0.61	0.57	0.56	0.28
$\bar{\varphi}_4$	-0.19	-0.16	-0.15	0.27
wall-clock time (hours)	437	210	198	222.5

Table 7.2: A summary of the results obtained using different parameters $\bar{\beta}$, β_{min}, and β_{max} in the one-shot routine compared to the nested optimization study. Cost function value and design variables are listed after 25000 one-shot iterations and state convergence. The wall-clock time is estimated for the 20-core workstation with 2 Intel® Xeon® processors of type E5-2670 v2, on which the routines are run, and is subject to coincidental parameters like the amount of jobs being processed. It excludes initial and final state and costate convergence for both nested and one-shot optimization. In comparison to the calculations listed in table 7.1, the finite difference parts of the gradient calculation are not parallelized here.

It can be seen that the last attempt with wider bounds (one-shot 3 in table 7.2) did not converge to the same optimum yet. For this test, convergence stalls at regular times. This can be seen from figure 7.9, where the cost function is shown for the different one-shot attempts. The red curve indeed stalls at intermediate values of the cost function. Further analysis shows that 93% of the one-shot iterations use the highest possible value of the Hessian, $B = \beta_{max}\delta_{ij}$. This leads to the smallest possible step in design space and thus a convergence

stall. With β_{min} as large as $5 \cdot 10^7$, they are even smaller than the finite difference steps in the in-parts adjoint gradient calculations. Therefore, they might give very misleading gradient estimations. Additionally, small steps near gradient discontinuities might lead to very large Hessian estimations that in turn give rise to even smaller steps. Therefore, a sufficiently restrictive upper Hessian bound β_{max} seems an important condition for convergence.

Looking into the 'one-shot 2' case in table 7.2, it can be seen that the same optimum is retrieved as the one found from the nested optimization study and the first one-shot case. However, looking at the blue curve of figure 7.9, one can see that convergence is very irregular. It seems rather shear luck that the optimization procedure does not diverge. It is clear that the increased maximal step size is the cause of this under-damped oscillatory convergence. Convergence would greatly benefit here from the use of a globalization approach. In conclusion, the Hessian bounds can thus not be relaxed much more without tolerating further convergence irregularities.

In regard to wall-clock times, it can be seen from table 7.2 that the grid deformation method turned the balance in favour of the one-shot method. Indeed, while in table 7.1 the one-shot method needed parallelization of the finite-difference gradient part to obtain a speed-up, this is no longer necessary when the grid deformation method is used. Moreover, while performing 5 times as much iterations now in comparison to the one-shot trial without grid deformation and simultaneously dropping the parallelization, the wall-clock time only doubled going from the first one-shot trial in table 7.1 to the one-shot runs in table 7.2.

Conclusions

A field-aligned grid deformation method is introduced that efficiently accelerates design steps in the one-shot method. Additionally, an adjusted test case is set-up to study one-shot convergence behaviour in absence of the gradient discontinuities introduced by the evitable choice of outer grid boundary. Indeed, the remaining gradient discontinuities are much smaller, but they do not disappear. In a study that attempts different parameters for the design step calculation, it is found

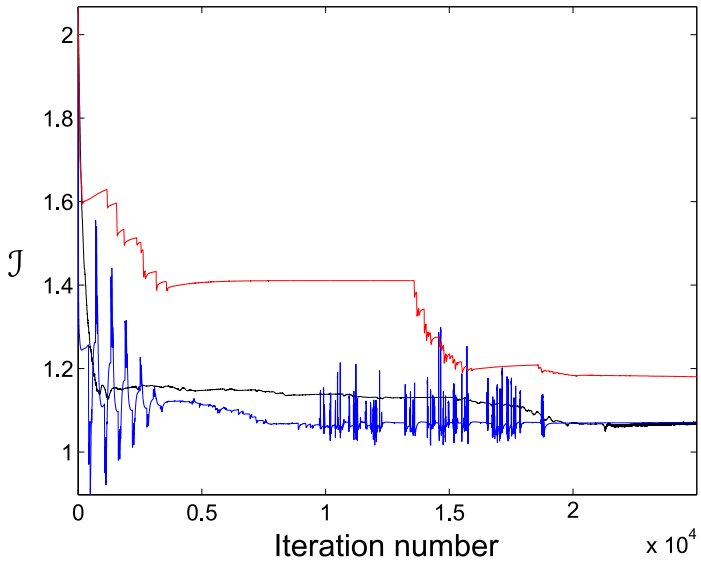


Figure 7.9: The evolution of the cost function value during one-shot optimization using different values $\bar{\beta}$, β_{min}, and β_{max} in the one-shot routine. The values for the different cases, one-shot 1 (black curve), one-shot 2 (blue curve), and one-shot 3 (red curve) are summarized in table 7.2.

that convergence is additionally very dependent on these parameters. To find parameter-independent convergence behaviour, a globalization procedure such as the one proposed by Hamdi and Griewank [66] might be considered. Using a line search procedure, the influence of the gradient discontinuities might additionally be eliminated. Therefore, the application of such a globalized one-shot procedure is the subject of the next section. Since the existing theory on one-shot globalization was derived for use in combination with algorithmic differentiation, the main focus of the next section will be on adapting this globalization strategy for its practical and efficient use in the present (in-parts) continuous adjoint framework. Additionally, a more sophisticated approach to Hessian estimation will be proposed that is less sensitive to the sporadic occurrence of gradient discontinuities.

7.4 A novel one-shot strategy with a practical augmented Lagrangian globalization

In the nested approach to optimal magnetic divertor design, globalization was achieved through a line search strategy that enforces the strong Wolfe conditions [12]. It is clear that in combination with one-shot optimization, these methods need revisiting. Indeed, looking solely at the objective function data in line searches is misleading since state and costate are not converged in each optimization step. In section 7.4.1, the doubly augmented Lagrangian method of Hamdi and Griewank [65, 66] is considered for this purpose. This requires introducing general notions on fixed-point solvers first, along with their use in optimization. After summarizing their methodology, remaining challenges for practical application are indicated. Throughout this section, a discrete setting is supposed for the model equations, since the original theory of Hamdi and Griewank is performed in a discrete setting. For ease of notation, the subscript $(\cdot)_h$ introduced in section 2.1.2 to indicate discretized variables will not be explicitly written. Nevertheless, the methods could in principle entirely be introduced in functional spaces. The reader is referred to Kaland et al. [81, 82] for the extensive treatment of the convergence theory in functional spaces. Next, starting from this sound theoretical basis, an adjusted strategy for globalized one-shot optimization is derived in sections 7.4.2 and 7.4.3 that aims at reducing implementation efforts and increasing the efficiency. Furthermore, a robust BFGS approach to Hessian estimation for one-shot optimization is elaborated in section 7.4.4. Recall from the previous section that curvature bounds were found to be important for the simplified Hessian calculation of Ref. [69]. Therefore, a strategy will be proposed here to include curvature bounds on the BFGS Hessian estimate. Finally, in section 7.5, the novel one-shot method is applied to the unconstrained optimization problem.

7.4.1 One-shot optimization using a doubly augmented Lagrangian function

Optimization with fixed-point solvers

The essential assumption of the fixed-point theory is that the model equations $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) = 0$ can be transformed in a fixed-point iteration form

$$\boldsymbol{q} = \boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}), \tag{7.18}$$

where the Jacobian $\partial_{\boldsymbol{q}}\boldsymbol{G}$ should have a spectral radius $\rho < 1$, in order to have a contractive iteration procedure $\boldsymbol{q}_{k+1} = \boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}_k)$ that leads to a feasible solution $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{\varphi})$. Remark that the fixed-point solver $\boldsymbol{G}$ can easily be linked to the model contraint $\boldsymbol{c}$ by the relation

$$\boldsymbol{c} = A(\boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}) - \boldsymbol{q}),$$

where A is given by the solution procedure. For a direct Newton solver, $A = -\partial_{\boldsymbol{q}}\boldsymbol{c}$. In the plasma edge code, a pseudo-time stepping scheme is used to iterate the equations $\frac{\partial \boldsymbol{q}}{\partial t} = \boldsymbol{c}$ to steady state. One could roughly say that this would correspond to $A = -1/\omega \left(\partial_{\boldsymbol{q}}\boldsymbol{c} - \mathbb{I}/\Delta t\right)$, with $\mathbb{I}$ an identity matrix of appropriate dimensionality and ω a constant relaxation factor. In reality, the state equations are solved with a segregated solver that gives rise to an inexact approximation of the latter expression, as discussed in section 3.3.4.

Using the fixed-point iteration form of the discrete model equations, the KKT conditions can be derived from the Lagrangian [65]

$$\mathcal{L}_G(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{q}) + (\boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}) - \boldsymbol{q})^\top \mathbf{y}^* = \mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) - \boldsymbol{q}^\top \mathbf{y}^*, \tag{7.19}$$

with $\mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*)$ the so-called *shifted Lagrangian*. The KKT-conditions then read

$$\begin{cases} \bar{\boldsymbol{q}} = \boldsymbol{G}(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}) & \text{State} \\ \bar{\mathbf{y}}^* = \nabla_{\boldsymbol{q}}\mathcal{N} = \partial_{\boldsymbol{q}}\mathcal{I}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top + \partial_{\boldsymbol{q}}\boldsymbol{G}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top \bar{\mathbf{y}}^* & \text{Adjoint} \\ 0 = \nabla_{\boldsymbol{\varphi}}\mathcal{N} = \partial_{\boldsymbol{\varphi}}\mathcal{I}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top + \partial_{\boldsymbol{\varphi}}\boldsymbol{G}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top \bar{\mathbf{y}}^* & \text{Design} \end{cases}. \tag{7.20}$$

In comparison to the KKT conditions based on the model equations $\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) = 0$,

$$\begin{cases} \nabla_{\boldsymbol{q}^*}\mathcal{L} = \boldsymbol{c}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right) = 0 & \text{State} \\ \nabla_{\boldsymbol{q}}\mathcal{L} = \partial_{\boldsymbol{q}}\mathcal{I}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top + \partial_{\boldsymbol{q}}\boldsymbol{c}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top \bar{\boldsymbol{q}}^* = 0 & \text{Adjoint} \\ \nabla_{\boldsymbol{\varphi}}\mathcal{L} = \partial_{\boldsymbol{\varphi}}\mathcal{I}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top + \partial_{\boldsymbol{\varphi}}\boldsymbol{c}\left(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}\right)^\top \bar{\boldsymbol{q}}^* = 0 & \text{Design} \end{cases},$$

the equations (7.20) give rise to slightly different adjoint variables $\bar{\mathbf{y}}^* = A^\top \bar{\boldsymbol{q}}^*$ that are found using the adjoint iterator [62]

$$\mathbf{y}^*_{k+1} = \nabla_{\boldsymbol{q}} \mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}_k, \mathbf{y}^*_k). \tag{7.21}$$

If iterations of $\boldsymbol{q}_{k+1} = \boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}_k)$ are used to converge the state $\boldsymbol{q}$ and the adjoint iterations $\mathbf{y}^*_{k+1} = \nabla_{\boldsymbol{q}} \mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}_k, \mathbf{y}^*_k)$ are used to converge the adjoint variables, the gradient of the reduced objective $\nabla \hat{\mathcal{J}}$ equals the design sensitivity $\nabla_{\boldsymbol{\varphi}} \mathcal{N}(\boldsymbol{\varphi}, \bar{\boldsymbol{q}}, \bar{\mathbf{y}}^*)$. Note that this framework is the natural representation of gradient calculations if backward mode AD is used. Indeed, the variables $\mathbf{y}^*_k = d\mathcal{J}/d\boldsymbol{q}_k$ correspond to the intermediate variables that are used in the backward swipe of AD to evaluate the objective gradient $\nabla \hat{\mathcal{J}}$ with the chain rule [63]. However, these type of gradient calculations are memory-intensive since all forward state variables $\boldsymbol{q}_k$ have to be kept to evaluate the adjoint iterates $\mathbf{y}^*_{k+1}$ with equation (7.21). Therefore, if a steady-state solution is sought, one might rather use a technique called *reverse accumulation* that replaces $\boldsymbol{q}_k$ in the evaluation of $\mathbf{y}^*_{k+1}$ by the final value of the state variables $\boldsymbol{q}_n = \boldsymbol{q}(\boldsymbol{\varphi})$ [31]. Alternatively, the earlier introduced piggyback iterations might be used to converge the state and adjoint equations simultaneously [64].

In this fixed-point framework, it can easily be seen that the adjoint iterator has the same spectral radius $\|\partial_{\boldsymbol{q}\mathbf{y}^*} \mathcal{N}\| = \|\partial_{\boldsymbol{q}} \boldsymbol{G}\|$ as the forward equation, which means that primal and adjoint have the same contraction rate. Since the adjoint equation depends on the state $\boldsymbol{q}$, the adjoint in general lags somewhat behind, a phenomenon called *dual retardation*. In Ref. [66], it was proven that asymptotically, for very demanding convergence criteria, the dual retardation becomes smaller than one. That is, the adjoint residuals catch up with the state residuals.

The essential last step is now the extension to one-shot optimization. In the next section, the convergence theory of Hamdi and Griewank is presented for a Jacobi-type one-shot iteration [65]. In each iteration, one step on state, adjoint, and design equation is performed as

$$\begin{bmatrix} \boldsymbol{q}_{k+1} \\ \mathbf{y}^*_{k+1} \\ \boldsymbol{\varphi}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}(\boldsymbol{\varphi}_k, \boldsymbol{q}_k) \\ \partial_{\boldsymbol{q}} \mathcal{N}(\boldsymbol{\varphi}_k, \boldsymbol{q}_k, \mathbf{y}^*_k)^\top \\ \boldsymbol{\varphi}_k - B_k^{-1} \partial_{\boldsymbol{\varphi}} \mathcal{N}(\boldsymbol{\varphi}_k, \boldsymbol{q}_k, \mathbf{y}^*_k)^\top \end{bmatrix}. \tag{7.22}$$

For a specific choice of preconditioner B, convergence can then be shown if the design space is assumed locally convex. Additionally, since the convergence proof is based on obtaining descent on an augmented Lagrangian type function, the use of a globalization approach that uses the same merit function is a quite natural extension elaborated in Ref. [66].

Doubly augmented Lagrangian theory

The one-shot method of Hamdi and Griewank looks for descent on the doubly augmented Lagrangian function

$$L_G^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \frac{\alpha_G}{2}\|\boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}) - \boldsymbol{q}\|^2 + \frac{\beta_G}{2}\|\nabla_{\boldsymbol{q}}\mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) - \mathbf{y}^*\|^2 + \mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) - \boldsymbol{q}^\top \mathbf{y}^*, \tag{7.23}$$

with α_G and β_G positive weighting coefficients for the forward and dual feasibility penalties, respectively. It was proven in [65] to be an exact penalty function if the weighting coefficients obey the inequality

$$\alpha_G \beta_G (1-\rho)^2 > 1 + \beta_G \theta \quad \text{, with} \qquad \theta = \|\partial_{\boldsymbol{qq}}\mathcal{N}\|. \tag{7.24}$$

This means that minimizing the augmented Lagrangian (7.23) yields the same solution as those given by the KKT system of the original problem. Furthermore, it is shown that the step increment vector

$$\boldsymbol{s}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \begin{bmatrix} \Delta \boldsymbol{q} = \boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}) - \boldsymbol{q} \\ \Delta \mathbf{y}^* = \partial_{\boldsymbol{q}}\mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*)^\top - \mathbf{y}^* \\ \Delta \boldsymbol{\varphi} = -B^{-1}\partial_{\boldsymbol{\varphi}}\mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*)^\top \end{bmatrix}, \tag{7.25}$$

is a descent direction of the doubly augmented Lagrangian, if the stronger condition

$$\sqrt{\alpha_G \beta_G}(1-\rho) > 1 + \frac{\beta_G}{2}\theta \tag{7.26}$$

is met and if a sufficiently large design space preconditioner B is chosen.

To prove this last property, the gradient of the augmented Lagrangian is decomposed in [65] as

$$\nabla L_G^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = -M\boldsymbol{s}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*), \text{where} \tag{7.27}$$

$$M = \begin{bmatrix} \alpha_G \Delta \boldsymbol{G}_{\boldsymbol{q}}^\top & -\mathbb{I} - \beta_G \partial_{\boldsymbol{qq}}\mathcal{N} & 0 \\ -\mathbb{I} & \beta_G \Delta \boldsymbol{G}_{\boldsymbol{q}} & 0 \\ -\alpha_G \partial_{\boldsymbol{\varphi}} G^\top & -\beta_G \partial_{\boldsymbol{q\varphi}}\mathcal{N}^\top & B \end{bmatrix}, \text{and} \quad \Delta \boldsymbol{G}_{\boldsymbol{q}} = \mathbb{I} - \partial_{\boldsymbol{q}}\boldsymbol{G}.$$

Requiring the step $\boldsymbol{s}$ to yield descent on L_G^a corresponds to requiring that

$$\boldsymbol{s}^\top \nabla L_G^a = -\boldsymbol{s}^\top M \boldsymbol{s} = -\boldsymbol{s}^\top \frac{M + M^\top}{2} \boldsymbol{s} < 0, \tag{7.28}$$

or, equivalently, that M or $(M + M^\top)/2$ is positive definite. Further elaboration shows that it is sufficient to choose B large enough, and

$$\alpha_G \beta_G \Delta \bar{\boldsymbol{G}}_{\boldsymbol{q}} \succ \left(\mathbb{I} + \frac{\beta_G}{2} \partial_{\boldsymbol{qq}} \mathcal{N} \right) \left(\Delta \bar{\boldsymbol{G}}_{\boldsymbol{q}} \right)^{-1} \left(\mathbb{I} + \frac{\beta_G}{2} \partial_{\boldsymbol{qq}} \mathcal{N} \right), \tag{7.29}$$

where $\Delta \bar{\boldsymbol{G}}_{\boldsymbol{q}} = 1/2 \left(\Delta \boldsymbol{G}_{\boldsymbol{q}} + \Delta \boldsymbol{G}_{\boldsymbol{q}}^\top \right)$. Condition (7.29) is then shown to be implied by equation (7.26).

Moreover, in [66] a sufficient condition for B was derived. Using large parameters α_G and β_G, equation (7.29) can easily be satisfied. However, increasing the penalties α_G and β_G leads to reduced design steps $\Delta \boldsymbol{\varphi}$ and therefore slower convergence of the optimization procedure. The penalties should therefore be chosen as small as possible to obtain a fast convergence of the one-shot iterations. Using (7.26) as equality and minimizing α_G leads to the choice of weighting coefficients

$$\alpha_G = \frac{2\theta}{(1-\rho)^2} \qquad \text{and} \qquad \beta_G = \frac{2}{\theta}. \tag{7.30}$$

For the design preconditioner B, it is pointed out in [66] that $B \approx \partial_{\boldsymbol{\varphi\varphi}} L_G^a$, which turns to an equality at primal and dual feasibility. A quasi-Newton BFGS update strategy was therefore proposed where

$$B \Delta \boldsymbol{\varphi} \approx \nabla_{\boldsymbol{\varphi}} L_G^a(\boldsymbol{\varphi} + \Delta \boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) - \nabla_{\boldsymbol{\varphi}} L_G^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) \tag{7.31}$$

is employed as a secant equation for the Hessian update, with

$$\partial_{\boldsymbol{\varphi}} L_G^a = \alpha_G \Delta \boldsymbol{q}^\top \partial_{\boldsymbol{\varphi}} \boldsymbol{G} + \beta_G \Delta \mathbf{y}^{*\top} \partial_{\boldsymbol{q\varphi}} \mathcal{N} + \partial_{\boldsymbol{\varphi}} \mathcal{N}. \tag{7.32}$$

Furthermore, a set of globalization strategies is listed that look for descent on L_G^a in the direction $\boldsymbol{s}$. Global convergence could then be proven under reasonable assumptions.

Discussion

The procedure of Hamdi and Griewank is the first one-shot procedure that guarantees convergence. However, implementation might be more cumbersome than the classical ad-hoc one-shot methods and the method might not be the fastest in terms of CPU time. In view of implementation efforts, especially the need of a second order adjoint (SOA) for the evaluation of $\partial_{\boldsymbol{q}\boldsymbol{\varphi}}\mathcal{N}^\top\Delta\mathbf{y}^*$ in (7.32) is a drawback. In particular in a continuous adjoint framework, the additional derivation and implementation of this second derivative are impractical. Furthermore, it adds two new equations to be solved to the iterative optimization procedure, namely the perturbed forward and the SOA equation, having also significant repercussions on computational cost. Alternatively, a finite difference approximation for this term is proposed in [66]. Also with this approach at least one additional state calculation is needed in each one-shot iteration. A solution to this problem is offered in the first part of section 7.4.2.

Secondly, the parameters derived are sufficient for convergence but not strictly necessary, so that smaller penalties might still be adequate and may lead to accelerated convergence. Also, it should be noted that the choice of an appropriate norm for the feasibility penalties in equation (7.23) is not straightforward. Especially in application of the approach to a problem governed by a system of PDE's, it is clear that using a regular 2-norm might lead to unwanted results. Indeed, depending on the units or scaling of individual equations, the convergence of the one-shot approach might then be significantly altered. Especially in nuclear fusion applications, where densities and temperatures are not in everyday ranges, one equation (e.g. ion continuity equation) in $\|\boldsymbol{G}(\boldsymbol{\varphi},\boldsymbol{q})-\boldsymbol{q}\|$ might dominate all other equations so that the feasibility penalty of other equations ends up in the range of numerical error. Additionally, line searches on the augmented Lagrangian function are quite expensive, since they require design, state, and costate evaluations. These arguments motivate a further reformulation of the feasibility penalty in section 7.4.2.

Thirdly, the evaluation of $\nabla_{\boldsymbol{\varphi}}L_G^a(\boldsymbol{\varphi}+\Delta\boldsymbol{\varphi},\boldsymbol{q},\mathbf{y}^*)$ needed in the Hessian update requires an additional evaluation of the design gradient. As appeared in the previous section, the evaluation of the design gradient has a dominant contribution

to the total computational cost. In section 7.4.4, a more convenient alternative will be offered, based on theoretical considerations.

7.4.2 Derivation of a practical augmented Lagrangian merit function

Considerations on the adjoint feasibility penalty

As discussed in section 7.4.1, the adjoint feasibility penalty in the doubly augmented Lagrangian (7.23) complicates implementation and slows down the one-shot optimization. Therefore, it may be worthwhile questioning the necessity of this term. In several applications, the θ occurring in the "optimal" penalty parameters (7.30) might well be approximated by $\theta \approx 1$ [105]. It is clear that for the slowly converging fixed-point solvers under consideration (ρ close to 1), α_G is several orders of magnitude larger than β_G. Additionally, the dynamics of the adjoint residuals often correlate very well with those of the state residuals. These arguments intuitively suggest that dropping the adjoint feasibility penalty might well be reasonable. Moreover, a similar argument may be raised as for piggyback iterations, in which bounded dual retardation is found [64]. Because of the typically observed bounded retardation, we assume here that the ratio of dual and primal updates is bounded by a coefficient L as $\|\Delta \mathbf{y}^*\| \leqslant L \|\Delta \boldsymbol{q}\|$.

Using the above arguments, we derive under which conditions the step $\boldsymbol{s_1}$ is a descent direction of the alternative augmented Lagrangian

$$L_S^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \frac{\alpha_G}{2} \|\boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}) - \boldsymbol{q}\|^2 + \mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) - {\mathbf{y}^*}^\top \boldsymbol{q}, \tag{7.33}$$

which only has a single state penalty and with

$$\boldsymbol{s_1}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \begin{bmatrix} \Delta \boldsymbol{q} = \boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}) - \boldsymbol{q} \\ \Delta \mathbf{y}^* = \partial_{\boldsymbol{q}} \mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*)^\top - \mathbf{y}^* \\ \Delta \boldsymbol{\varphi} = -B^{-1} \partial_{\boldsymbol{\varphi}} L_S^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*)^\top \end{bmatrix}, \tag{7.34}$$

where $\partial_{\boldsymbol{\varphi}} L_S^a$ is used instead of $\partial_{\boldsymbol{\varphi}} \mathcal{N}$ in the design step calculation to avoid a priori any possible inconsistencies between descent on cost function and augmented Lagrangian function in the line search algorithm (further explained in section 7.4.3).

Similar to equation (7.27), requiring descent on L_S^a for a one-shot iteration corresponds to

$$\begin{aligned} &\boldsymbol{s_1}^\top \nabla L_S^a \approx -\boldsymbol{s_1}^\top M_1 \boldsymbol{s_1} < 0, \text{with} \\ &M_1 = \begin{bmatrix} \alpha_G \Delta \boldsymbol{G_q}^\top & -\mathbb{I} & 0 \\ -\mathbb{I} & 0 & 0 \\ 0 & 0 & B \end{bmatrix}. \end{aligned} \tag{7.35}$$

Remind that $\Delta \boldsymbol{G_q}$ is defined cfr. equation (7.27). For a general vector $\boldsymbol{v} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]^\top$, one can observe that

$$\begin{aligned} \boldsymbol{v}^\top M_1 \boldsymbol{v} &= \alpha_G \boldsymbol{v}_1^\top \Delta \boldsymbol{G_q}^\top \boldsymbol{v}_1 - 2\boldsymbol{v}_2^\top \boldsymbol{v}_1 + \boldsymbol{v}_3^\top B \boldsymbol{v}_3, \\ &\geqslant \alpha_G (1-\rho) \|\boldsymbol{v}_1\|^2 - 2\|\boldsymbol{v}_1\| \|\boldsymbol{v}_2\| + \lambda_{\min}(B) \|\boldsymbol{v}_3\|^2, \end{aligned}$$

with $\lambda_{\min}(B)$ the smallest eigenvalue of B. Using the bounded retardation of the adjoint residuals, descent is implied by

$$\left(\alpha_G (1-\rho) - 2L\right) \|\Delta \boldsymbol{q}\|^2 + \lambda_{\min}(B) \|\Delta \boldsymbol{\varphi}\|^2 > 0.$$

It is thus sufficient to choose the design preconditioner B to be positive definite and

$$\alpha_G > \frac{2L}{1-\rho}. \tag{7.36}$$

Since coupled state-costate-design iterations continuously descend on the augmented Lagrangian function L_S^a, the one-shot method will converge to the optimum $(\bar{\boldsymbol{q}}, \bar{\boldsymbol{y}}^*, \bar{\boldsymbol{\varphi}})$. In conclusion, the adjoint residuals in the doubly augmented Lagrangian can thus be disregarded. As such, the approach is simplified considerably.

A practical augmented Lagrangian merit function

In section 7.4.1, it was also pointed out that the choice of norm in the penalty contributions is not evident, as the different PDE's have different units and scales. Indeed, state variables of continuity, momentum or energy equation have all different dimension and different orders of magnitude. It seems therefore more appropriate to penalize state feasibility using the normalized residuals of the discretized equations instead of using the state step $\Delta \boldsymbol{q}$. In many codes they serve

as convergence monitor and are therefore at hand. This gives an augmented Lagrangian function of the form

$$L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \frac{\alpha}{2} \sum_{i=1}^{n_E} \| \boldsymbol{c_i}(\boldsymbol{\varphi}, \boldsymbol{q}) / s_{c,i} \|^2 + \mathcal{L}_G(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*), \tag{7.37}$$

with n_E then the number of equations, $s_{c,i}$ a normalization factor and i an index denoting the corresponding equation. When the equations are normalized in advance one can write this as

$$L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \frac{\alpha}{2} \|\boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q})\|^2 + \mathcal{L}_G(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*). \tag{7.38}$$

It may be interesting to see whether also in this form, an α and B can be found so that a coupled state-design step

$$\boldsymbol{s_2}\,(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) = \begin{bmatrix} \Delta \boldsymbol{q} = \boldsymbol{G}(\boldsymbol{\varphi}, \boldsymbol{q}) - \boldsymbol{q} \\ \Delta \mathbf{y}^* = \partial_{\boldsymbol{q}} \mathcal{N}(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*)^\top - \mathbf{y}^* \\ \Delta \boldsymbol{\varphi} = -B^{-1} \partial_{\boldsymbol{\varphi}} L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*)^\top \end{bmatrix}, \tag{7.39}$$

offers descent in L^a. The only change with respect to the derivation in section 7.4.2 is of course the feasibility penalty term. Using the relation $\boldsymbol{c} = A\Delta\boldsymbol{q}$, the descent condition can be written similar to equation (7.35), namely

$$\begin{aligned} &\boldsymbol{s_2}^\top \nabla L^a \approx -\boldsymbol{s_2}^\top M_2 \boldsymbol{s_2} < 0, \text{with} \\ &M_2 = \begin{bmatrix} -\alpha\, \partial_{\boldsymbol{q}} \boldsymbol{c}^\top A & -\mathbb{I} & 0 \\ -\mathbb{I} & 0 & 0 \\ 0 & 0 & B \end{bmatrix}, \end{aligned} \tag{7.40}$$

and where the only difference between M_2 and M_1 is clearly the $-\alpha\, \partial_{\boldsymbol{q}} \boldsymbol{c}^\top A$ term in the matrix M_2. Therefore, a lower bound is sought for this term. For this purpose, one should first notice that

$$\partial_{\boldsymbol{q}} \boldsymbol{c} \Delta \boldsymbol{q} = -A \Delta \boldsymbol{G_q} \Delta \boldsymbol{q} + \partial_{\boldsymbol{q}} A \Delta \boldsymbol{q} \Delta \boldsymbol{q} \approx -A \Delta \boldsymbol{G_q} \Delta \boldsymbol{q},$$

where the latter is a good approximation since $\Delta \boldsymbol{q}$ tends to zero during iterations and $\partial_{\boldsymbol{q}} A$, typically approximating $-\partial_{\boldsymbol{qq}} \boldsymbol{c}$, is a higher order term. Using this approximation, one finds

$$\begin{aligned} -\alpha \Delta \boldsymbol{q}^\top \partial_{\boldsymbol{q}} \boldsymbol{c}^\top A \Delta \boldsymbol{q} &\approx \alpha \Delta \boldsymbol{q}^\top \Delta \boldsymbol{G}_{\boldsymbol{q}}^\top A^\top A \Delta \boldsymbol{q} \\ &\geqslant \alpha a^2 (1 - \rho) \|\Delta \boldsymbol{q}\|^2, \end{aligned} \tag{7.41}$$

where a^2 is the lowest eigenvalue of $A^\top A$. Combining the lower bound for the new term in (7.41) to a similar reasoning as the previous derivation, the condition

$$\left(\alpha a^2(1-\rho) - 2L\right) \|\Delta \boldsymbol{q}\|^2 + \lambda_{\min}(B)\|\Delta \boldsymbol{\varphi}\|^2 > 0.$$

is found to be a sufficient condition for descent on L^a. Again, one concludes that choosing a large positive preconditioner B and a large enough penalty parameter α to meet the condition

$$\alpha > \frac{2L}{a^2(1-\rho)} \tag{7.42}$$

will lead to decrease in L^a for one-shot iterations and therefore ultimately to a solution of the KKT-system (7.20).

7.4.3 A simple line search procedure for globalization

One can now use the novel augmented Lagrangian formulation (7.37) as a merit function to globalize the one-shot optimization procedure. This will extend the local convergence to convergence from any initial design point. For this purpose, a simple line search algorithm based on the augmented Lagrangian function L^a is introduced. A design step $\Delta\boldsymbol{\varphi}$ will only be accepted if it obeys the Armijo condition

$$L^a(\boldsymbol{\varphi} + \Delta\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) < L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) + \gamma \nabla_{\boldsymbol{\varphi}} L^{a\top} \Delta\boldsymbol{\varphi}, \tag{7.43}$$

with $\gamma \in (0,1)$. Otherwise, the step $\Delta\boldsymbol{\varphi}$ is reduced with a constant fraction (a procedure commonly known as backtracking) until the Armijo condition is satisfied. In this perspective, the augmented Lagrangian formulation L^a defined in (7.37) has clear advantages, since during line search, there is no need for an adjoint update in this case. In contrast, the original approach of Hamdi and Griewank requires also calculating the adjoint update for the dual feasibility penalty. Assuming that the Lagrangian $\mathcal{L}_G$ in fixed-point form can be substituted in the augmented Lagrangian L^a by the Lagrangian

$$\mathcal{L} = \mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{q}) + \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q})^\top \boldsymbol{q}^*$$

in residual form, the line search cost can be further reduced. Then, only the residuals of the state equations are needed in each line search iteration. Therefore, one avoids the solution of the system $\Delta\boldsymbol{q} = A^{-1}\boldsymbol{c}$ as well.

7.4.4 A robust approach to Hessian estimation

The choice of a suffiently large design preconditioner B is clearly essential for the convergence of the one-shot iterations. On the other hand, superlinear convergence is desired, while additional efforts to calculate B should be minimal. The BFGS approach is therefore a good candidate. However, as concluded in section 7.4.1, the choice $B \approx \partial_{\boldsymbol{\varphi}\boldsymbol{\varphi}} L_G^a$ suggested by Hamdi and Griewank requires an additional design sensitivity evaluation. Therefore, the approximation is reconsidered. As will appear from what follows, a consistent derivation of a design preconditioner B based on the minimization of a second-order expansion of L^a will lead to a different (and cheaper) result than the tangent rule (7.31) proposed by Hamdi and Griewank.

To find the best design preconditioner B, the design step

$$\Delta\boldsymbol{\varphi} = -B^{-1}\nabla_{\boldsymbol{\varphi}} L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) \tag{7.44}$$

is sought that maximizes the descent on a quadratic approximation of L^a. This $\Delta\boldsymbol{\varphi}$ solves the minimization problem

$$\min_{\Delta\boldsymbol{\varphi}} \boldsymbol{s}^\top \nabla L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) + \frac{1}{2}\boldsymbol{s}^\top \nabla^2 L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*)\boldsymbol{s}. \tag{7.45}$$

Keeping only the terms that depend on $\Delta\boldsymbol{\varphi}$, the minimization problem reduces to

$$\min_{\Delta\boldsymbol{\varphi}} E(\Delta\boldsymbol{\varphi}),$$

with

$$E(\Delta\boldsymbol{\varphi}) = \frac{1}{2}\Delta\boldsymbol{\varphi}^\top \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}} L^a \Delta\boldsymbol{\varphi} + \Delta\boldsymbol{\varphi}^\top \left(\nabla_{\boldsymbol{\varphi}} L^a + \nabla_{\boldsymbol{\varphi}\boldsymbol{q}} L^a \Delta\boldsymbol{q} + \nabla_{\boldsymbol{\varphi}\mathbf{y}^*} L^a \Delta\mathbf{y}^*\right).$$

To find the minimum, we impose that the derivative of E with respect to $\Delta\boldsymbol{\varphi}$ equals zero:

$$\nabla_{\boldsymbol{\varphi}} L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*) + \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}} L^a \Delta\boldsymbol{\varphi} + \nabla_{\boldsymbol{\varphi}\boldsymbol{q}} L^a \Delta\boldsymbol{q} + \nabla_{\boldsymbol{\varphi}\mathbf{y}^*} L^a \Delta\mathbf{y}^* = 0.$$

Eliminating the design gradient $\nabla_{\boldsymbol{\varphi}} L^a$ using the design step relation (7.44) then leads to

$$B\Delta\boldsymbol{\varphi} = \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}} L^a \Delta\boldsymbol{\varphi} + \nabla_{\boldsymbol{\varphi}\boldsymbol{q}} L^a \Delta\boldsymbol{q} + \nabla_{\boldsymbol{\varphi}\mathbf{y}^*} L^a \Delta\mathbf{y}^*.$$

From this equality, we may now derive a secant equation for Hessian updates:

$$B\Delta\boldsymbol{\varphi} \approx \nabla_{\boldsymbol{\varphi}} L^a(\boldsymbol{\varphi} + \Delta\boldsymbol{\varphi}, \boldsymbol{q} + \Delta\boldsymbol{q}, \mathbf{y}^* + \Delta\mathbf{y}^*) - \nabla_{\boldsymbol{\varphi}} L^a(\boldsymbol{\varphi}, \boldsymbol{q}, \mathbf{y}^*). \tag{7.46}$$

During one-shot optimization, we therefore use the BFGS update formula,

$$B_{k+1} = B_k - \frac{B_k \boldsymbol{s_k} \boldsymbol{s_k}^\top B_k}{\boldsymbol{s_k}^\top B_k \boldsymbol{s_k}} + \frac{\boldsymbol{y_k} \boldsymbol{y_k}^\top}{\boldsymbol{y_k}^\top \boldsymbol{s_k}},$$

with $\boldsymbol{s_k} = \boldsymbol{\varphi}_{k+1} - \boldsymbol{\varphi}_k$ and

$$\boldsymbol{y_k} = \nabla_{\boldsymbol{\varphi}} L^a(\boldsymbol{\varphi}_{k+1}, \boldsymbol{q}_{k+1}, \mathbf{y}^*_{k+1}) - \nabla_{\boldsymbol{\varphi}} L^a(\boldsymbol{\varphi}_k, \boldsymbol{q}_k, \mathbf{y}^*_k),$$

to update the design preconditioner B. If the convergence of state or adjoint residuals is noisy, one might decide to use other secant equations then equation (7.46), such as the secant equation (7.31). As argued by Hamdi and Griewank [66], these approximations becomes valid at primal and dual feasibility. Notice that this issue therefore only arises in one-shot approaches.

To safeguard the positive definiteness of the Hessian estimate B, Powell's damped update rule can be used. This update rule relaxes the Hessian update to guarantee that the directional curvature can maximally be decreased by a fraction γ_{min} [112]. The damped BFGS approach thus consists of applying the following rule: If $\boldsymbol{s_k}^\top \boldsymbol{y_k} < \gamma_{min}\, \boldsymbol{s_k}^\top B_k \boldsymbol{s_k}$, substitute $\boldsymbol{y_k}$ by $\tilde{\boldsymbol{y}}_{\boldsymbol{k}} = \theta \boldsymbol{y_k} + (1-\theta) B_k \boldsymbol{s_k}$, such that $\boldsymbol{s_k}^\top \tilde{\boldsymbol{y}}_{\boldsymbol{k}} = \gamma_{min}\, \boldsymbol{s_k}^\top B_k \boldsymbol{s_k}$. This is achieved when

$$\theta = \frac{(1-\gamma_{min}) \boldsymbol{s_k}^\top B_k \boldsymbol{s_k}}{\boldsymbol{s_k}^\top B_k \boldsymbol{s_k} - \boldsymbol{s_k}^\top \boldsymbol{y_k}}. \tag{7.47}$$

To make the Hessian estimation more robust, absolute bounds can be added on the curvature as in the simplified approach of Hazra et al. [69]. Indeed, one needs to avoid that the Hessian definiteness is decreased up to machine accuracy in successive iterations, resulting in unrealistically large design steps. Additionally, a safety margin should be kept, since Powell's trick is a rank-two update and only the definiteness of the Hessian in the step direction is monitored. Similarly, an unbounded increase of the Hessian causes numerical problems as well. As was discussed in section 7.3.3, too small design steps will cause the gradient differences $\boldsymbol{y_k}$ to be in the error margin of the gradients, which might lead to a

further misleading increase of the Hessian estimate. Especially when using (in-parts) finite difference sensitivity calculations, it seems reasonable to demand that the design steps are an order of magnitude bigger than the finite difference steps. Additionally, small steps near gradient discontinuities might lead to very large Hessian estimations that give rise to even smaller steps. For all these reasons, (7.47) is used to bound the relative decrease as well as the relative increase. The latter is then done when $\boldsymbol{s_k}^\top \boldsymbol{y_k} > \gamma_{max}\, \boldsymbol{s_k}^\top B_k \boldsymbol{s_k}$.

Similar to the simple method of Hazra et al. [69], absolute bounds β_{min} and β_{max} on the directional curvature can be imposed in addition. To this end, it is proposed here to modify Powell's trick as follows. The directional curvature of each step should be adjusted so that

$$\beta_{min} < \frac{\boldsymbol{s_k}^\top \boldsymbol{y_k}}{\boldsymbol{s_k}^\top \boldsymbol{s_k}} < \beta_{max}. \tag{7.48}$$

If any of the two above constraints is violated $\tilde{\boldsymbol{y}}_{\boldsymbol{k}}$ can be calculated from

$$\boldsymbol{s_k}^\top \tilde{\boldsymbol{y}}_{\boldsymbol{k}} = \beta \boldsymbol{s_k}^\top \boldsymbol{s_k}, \tag{7.49}$$

where β represents either of the two bounds β_{min} or β_{max}. Solving (7.49) for θ leads to

$$\theta = \frac{\boldsymbol{s_k}^\top \left(B_k - \beta\, \mathbb{I}\right) \boldsymbol{s_k}}{\boldsymbol{s_k}^\top B_k \boldsymbol{s_k} - \boldsymbol{s_k}^\top \boldsymbol{y_k}}. \tag{7.50}$$

Finally, it should be noted that in one-shot optimization a lot more updates are done to the Hessian than in regular optimization. It is therefore logical to relax the Hessian updates as well. This is achieved when θ is substituted by $\tilde{\theta} = \mu\theta$, with $\mu < 1$ the relaxation factor.

7.5 Application of the globalized one-shot method to the unconstrained test problem

With the novel one-shot methodology developed in section 7.4 at hand, application to magnetic divertor design can now be considered. First, some comments are made with respect to the practical implementation of the algorithm. Next, an improved approach to in-parts adjoint sensitivity calculation of the augmented

Lagrangian is introduced. The applied algorithm is then summarized in section 7.5.3. In the last part of this section, the results of the novel one-shot method are presented.

7.5.1 From theory to practice

It should be noted that only in the fixed-point iteration form, the adjoint equations are guaranteed to have the same contraction rate as the forward equations. Therefore, one-shot convergence can only be derived in this framework. In the more general continuous adjoint framework on which the adjoint plasma edge equations are based, the contraction rate of the adjoint solver depends on the choice of the discretization and the iteration procedure. However, if a consistent approach is used to implement the adjoint equations, one might retrieve this property for continuous adjoint methods. Looking back at the piggyback iterations after optimization in figure 7.8 (part III), it can indeed be seen that also for the continuous adjoint plasma equations, used in this work, this seems to hold. For sure, the presence of a consistent adjoint pressure-correction equation contributes to this consistency [40]. It is therefore assumed that the continuous adjoint framework can be reused.

In practice, descent is then sought on the partially reduced augmented Lagrangian

$$\widetilde{L}^a(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}}) = \frac{\alpha}{2} \sum_{i=1}^{n_E} \| \frac{\widetilde{\boldsymbol{c}}_{\mathrm{pe},\boldsymbol{i}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})}{s_{c,i}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})} \|_2^2 + \widetilde{\mathcal{L}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}}), \tag{7.51}$$

with $\widetilde{\mathcal{L}}$ the discretization of the partially reduced Lagrangian defined in Eq. (5.6). The Euclidian norm $\|\cdot\|_2$ is used here in the state constraint penalty. Recalling the definition of the partially reduced plasma edge transport equations $\widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) := \boldsymbol{c}_{\mathrm{pe}}(\boldsymbol{c}_{\mathrm{gg}}(q_{\mathrm{eq}}(\boldsymbol{\varphi})), \boldsymbol{q}_{\mathrm{pe}})$, $\widetilde{\boldsymbol{c}}_{\mathrm{pe},\boldsymbol{i}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})$ is defined as the i^{th} (discretized) partial differential equation in partially reduced form. For the plasma edge transport model used in this work, consisting of plasma continuity, parallel momentum, total internal energy, and neutral pressure diffusion equations, the number of equations $n_E = 4$. Next, it should be mentioned that although the derivations of section 7.4 are done for a Jacobi-type iteration of state, adjoint, and design

equations, Gauss-Seidel-type iterations will still be used. In practice, they are often found to converge much faster [23, 62].

It should also be noted that the Hessian updates for the selected results follow the methodology as discussed in section 7.4.4 but are based on the secant rule

$$B\Delta\boldsymbol{\varphi} \approx \nabla_{\boldsymbol{\varphi}}\widetilde{L}^a(\boldsymbol{\varphi}+\Delta\boldsymbol{\varphi}, \boldsymbol{q}+\Delta\boldsymbol{q}, \boldsymbol{q}^*) - \nabla_{\boldsymbol{\varphi}}\widetilde{L}^a(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*).$$

That is, in comparison to the secant rule that followed from theory, the variable $\boldsymbol{q}^*$ is kept constant. Both approximations have been tested but the choice was not observed to be of great importance for these cases. The final design step is relaxed by a further factor of two so that most first line search steps are accepted.

Finally, finding a suitable approach to estimate the parameters L and a^2 in the sufficient condition for the penalty parameter α (equation (7.42)) remains a challenge for future work. Furthermore, the condition itself might be too restrictive. For these reasons, the feasibility penalty parameter α is not chosen based on this sufficient condition. Rather, α was chosen experimentally by trying several values for a limited amount of iterations. Results for different values of α will be given.

7.5.2 Notes on sensitivity calculation

An accurate and fast evaluation of the design gradient $\nabla_{\boldsymbol{\varphi}}\widetilde{L}^a$ is essential for the success of the novel one-shot method. Similar to the in-parts adjoint approach of chapter 5, finite difference evaluations can be applied to directly approximate the directional derivatives of the augmented Lagrangian: $(\widetilde{L}^a)' := \partial_{\boldsymbol{\varphi}}\widetilde{L}^a\boldsymbol{\delta\varphi} \approx \Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{L}^a$. However, the finite difference steps should be small enough to avoid inconsistencies in line searches. It is obvious that especially the sensitivity of the squared terms in $\widetilde{L}^a$ will suffer from poor accuracy for small finite difference perturbations ε. Indeed, since only a finite amount of significant digits is used, the squaring operation will cause a loss of accuracy due to cancellation errors and will restrict the user to use rather large values for the finite difference perturbation ε. Therefore, a semi-analytical approach is proposed to eliminate the squaring operation. E.g., for the derivative of the feasibility penalty term this gives

$$\left(\frac{\alpha}{2}\sum_{i=1}^{n_E}\|\frac{\widetilde{\boldsymbol{c}}_{\mathrm{pe},\boldsymbol{i}}}{s_{c,i}}\|_2^2\right)' \approx \alpha\sum_{i=1}^{n_E}\frac{\widetilde{\boldsymbol{c}}_{\mathrm{pe},\boldsymbol{i}}{}^{\top}\Delta^{\varepsilon}_{\boldsymbol{\varphi}}\widetilde{\boldsymbol{c}}_{\mathrm{pe},\boldsymbol{i}}}{s_{c,i}^2} - \alpha\sum_{i=1}^{n_E}\|\frac{\widetilde{\boldsymbol{c}}_{\mathrm{pe},\boldsymbol{i}}}{s_{c,i}}\|_2^2\frac{\Delta^{\varepsilon}_{\boldsymbol{\varphi}}s_{c,i}}{s_{c,i}}. \tag{7.52}$$

It may be noticed that the scaling variables $s_{c,i}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})$ that make the residuals dimensionless depend indirectly on the control variables through the coordinate transformation $\boldsymbol{q}_{\mathrm{gg}}(\psi(\boldsymbol{\varphi}))$. This leads to the second term on the right-hand side of equation (7.52). A similar method is applied to other squared objective function terms of the original objective functional in (4.4). Because the cancellation error in the finite difference approximations can be significantly reduced in this way, the finite difference step can be decreased from $\varepsilon = \varepsilon_{\mathrm{m}}^{1/4}$ to $\varepsilon = \varepsilon_{\mathrm{m}}^{1/2}$. Finally, the minimal design step in line search is limited to 10 times the finite difference step size ε to avoid inconsistencies in line search.

7.5.3 Overview of the applied one-shot algorithm

The different steps in the applied one-shot algorithm are summarized here.

Algorithm 7.2. *A practical globalized one-shot algorithm for magnetic divertor design*

1. *Perform one iteration on the state solver of the plasma edge equations* $\widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{pe}) = 0$. *For the current model, this requires calculating sequentially the poloidal magnetic flux* ψ, *the metric coefficients of the deformed grid* $\boldsymbol{q}_{gg}$, *and performing a pseudo-time step of the plasma edge solver for the updated state variables* $\boldsymbol{q}_{pe,k+1}$.

2. *Perform one pseudo-time step in the segregated solver of the continuous adjoint plasma edge equations to obtain the updated adjoint variables* $\boldsymbol{q}^*_{pe,k+1}$.

3. *Evaluate the gradient of the augmented Lagrangian. This is done here by approximately evaluating the gradient* $\nabla_{\boldsymbol{\varphi}} \widetilde{L}^a$ *of the augmented Lagrangian* $\widetilde{L}^a$ *(equation* (7.51)*) with finite differences. To increase the accuracy of the finite difference evaluations, the semi-analytical approach illustrated in equation* (7.52) *is used.*

4. *Calculate the design preconditioner* B *from the robust BFGS strategy described in section 7.4.4 and calculate a search direction from*

$$\Delta\boldsymbol{\varphi}_k = -B^{-1}\nabla_{\boldsymbol{\varphi}} \widetilde{L}^a.$$

5. *Reduce the design step* $\Delta\boldsymbol{\varphi}_k$ *in a backtracking line search until the Armijo condition*

$$\widetilde{L}^a(\boldsymbol{\varphi}_k + \Delta\boldsymbol{\varphi}_k, \boldsymbol{q}_{pe,k+1}, \boldsymbol{q}^*_{pe,k+1}) < \widetilde{L}^a(\boldsymbol{\varphi}_k, \boldsymbol{q}_{pe,k+1}, \boldsymbol{q}^*_{pe,k+1}) + \gamma\Delta\boldsymbol{\varphi}_k^\top \nabla_{\boldsymbol{\varphi}}\widetilde{L}^a,$$

is met and evaluate the updated design variables $\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k + \Delta\boldsymbol{\varphi}_k$.

These steps are then repeated until design, state, and adjoint variables are converged to the solution of the KKT conditions. Remark that this algorithm has been tailored to meet the requirements of the current application and the existing continuous adjoint framework. Different variants of the proposed one-shot algorithm are possible that may use both continuous or discrete adjoint (AD) sensitivity calculations to suit a specific application.

7.5.4 Results

As explained, the penalty parameter α in the augmented Lagrangian function was determined experimentally. For $\alpha = 10^{-8}$, fast, yet steady convergence was found. After first discussing the results for this particular choice of α, the influence of a different choice of α on convergence will be addressed.

The residuals are shown in figure 7.10. After converging the initial plasma state and adjoint using piggybacking, apparently noisy residuals changes are found. Again, many of the sudden surges of the residuals coincide with remeshing steps. The design residuals (green curve) reveal that gradient discontinuities are still present. However, these sudden increases in the design gradient do not automatically lead to large design steps, since sufficient descent on the augmented Lagrangian is required.

Indeed, looking at the augmented Lagrangian function value throughout optimization in figure 7.11, one may see that it decreases almost monotonically, despite the gradient discontinuities. Only the remeshing stages induce peaks disturbing this monotonic decrease. One might observe here that the increases in state and adjoint residuals in figure 7.10 (e.g. around iteration 0 and 40.000) are compensated by decreases in objective functional. Furthermore, it can be seen that cost functional and augmented Lagrangian functional differ significantly when the state and adjoint residuals are high. Notice that the cost function and

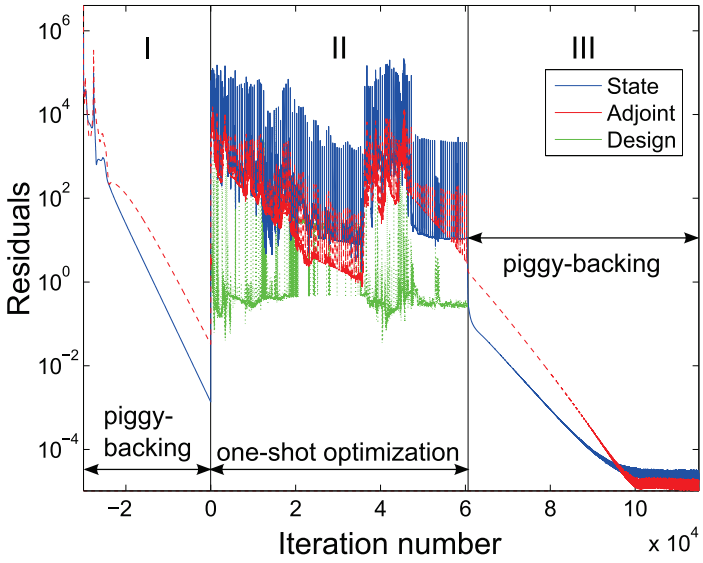


Figure 7.10: Convergence of the one-shot optimization procedure. State residuals are indicated with a solid blue line, adjoint residuals with a dashed red line, and design residuals with green dots. Piggy-backing iterations are used for feasibility convergence before (I) and after (III) one-shot optimization.

the augmented Lagrangian were not evaluated for the first 10000 iterations of the initial piggybacking stage. The difference is mainly attributed to term (2) of the Lagrangian

$$\widetilde{\mathcal{L}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}, \boldsymbol{q}^*_{\mathrm{pe}}) = \underbrace{\widetilde{\mathcal{J}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}})}_{\text{term (1)}} + \underbrace{\left\langle \boldsymbol{q}^*_{\mathrm{pe}}, \widetilde{\boldsymbol{c}}(\boldsymbol{\varphi}, \boldsymbol{q}_{\mathrm{pe}}) \right\rangle}_{\text{term (2)}}.$$

One may interpret this term as a linearized estimation of the effect that converging the state equation will have on the objective functional. Finally, it should be noted that further converging state and adjoint equations after optimization using piggybacking does not greatly change the cost functional, indicating that the one-shot procedure was sufficiently converged. Remark that full convergence of the one-shot optimization is not possible, since a continuous adjoint approach is

used. Also convergence of the state residuals in the one-shot iterations is not possible because of the presence of the minimally allowed design step, introduced in section 5.2.

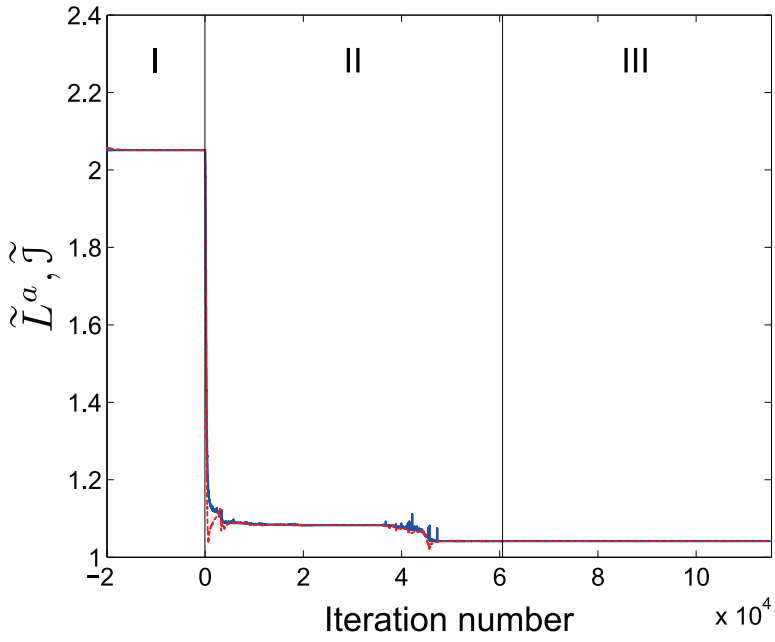


Figure 7.11: The augmented Lagrangian function $\widetilde{L}^a$ (solid blue line) and the cost function value $\widetilde{\mathcal{J}}$ (dashed red line) during one-shot and feasibility convergence with the novel one-shot method. The Roman numbers indicate the different regimes and correspond to the numbers in figure 7.10. The numbers I and III indicate piggybacking before and after one-shot optimization (II), respectively.

In comparison to the nested optimization method, very similar design variables and cost function value are retrieved. This can be seen in table 7.3 in the column "AL1". When increasing the penalty parameter α by a factor of 10, however, this optimum is no longer retrieved (see "AL2"). In that case, the optimization procedure suddenly stops at a higher cost function value. Further analysis indicates that convergence stalls at a gradient discontinuity, after which

the line search decreases the step size in each iteration up to the minimal value. Since design steps oscillate around the same design value, a local optimum seems to be created by the gradient discontinuity. If the bound β_{max} on the Hessian is decreased, convergence stalls at yet another cost function value (see "AL3").

	Nested	AL 1	AL 2	AL 3
α	\	$1 \cdot 10^{-8}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
β_{min}	\	10	10	10
β_{max}	\	$5 \cdot 10^{5}$	$5 \cdot 10^{5}$	$1 \cdot 10^{4}$
θ	\	0.02	0.02	0.02
γ_{min}	\	0.2	0.2	0.2
γ_{max}	\	10	10	10
$\hat{\mathfrak{J}}(\bar{\boldsymbol{\varphi}})$	1.08	1.07	1.36	1.18
$\bar{\varphi}_1$	0.76	0.75	0.68	0.74
$\bar{\varphi}_2$	0.07	0.12	0.39	0.40
$\bar{\varphi}_3$	0.61	0.61	0.40	0.18
$\bar{\varphi}_4$	-0.19	-0.22	0.37	0.43

Table 7.3: A summary of the results obtained using different parameters α, β_{min}, and β_{max} in the novel one-shot routine. The nested optimization study is shown for comparison. Final cost function value and design variables are listed.

While the simple one-shot method of section 7.2 lacks robust convergence, the current method will become slow if many inconsistencies of gradients are found because of the increased number of line search iterations. Furthermore, the method might converge to the local optima these discontinuities possibly create. Though increased robustness is found in terms of convergence towards these local optima, they are clearly suboptimal in terms of cost function decrease. It is clear that the step size plays a big role in the chance of getting stuck in such a local optimum. If big steps are taken, one might simply jump over them. Therefore, the augmented Lagrangian method converges to lower cost function values for smaller penalty factors α or lower Hessian bounds β_{max}. It is for the same reason that the nested optimization method is less sensitive to these local optima of the discrete cost function.

In conclusion, the novel one-shot method based on a singly-augmented Lagrangian cost function converges robustly, even in the presence of gradient discontinuities. Therefore, the method might serve as a practical and cheap alternative for the method of Hamdi and Griewank [66] in a number of one-shot optimization applications that encounter difficulties to converge in the absence of a line search globalization. In the current application, however, the method stalls at discrete local optima near gradient discontinuities. Furthermore, relative and absolute bounds on the BFGS estimation in combination with an update relaxation successfully avoid that sporadic discontinuities dominate the Hessian estimate. Nevertheless, given the noisy gradient, the accuracy of this estimate can be questioned.

Further research could therefore consider taking the preconditioner as a scaled identity matrix $B = \bar{\beta}\mathbb{I}$, turning the algorithm in a globalized steepest descent version of the novel one-shot method. Alternatively, efforts could be spent on increasing the smoothness of the code. Finally, increasing the Tikhonov regularization might improve the conditioning of the problem and remove some local optima.

7.6 One-shot optimization for constrained magnetic divertor design

Since the original problem examined in this work is a constrained optimization problem, a short outlook is given here for a strategy that includes the constraints in a one-shot approach. Especially for inequality-constrained optimization, few literature is available. In the thesis of Gherman [55], the approach of Hazra et al. [69] is extended to include a single inequality state constraint on aerodynamic lift. However, since it is assumed a priori that this state constraint is active at the optimum, it is de facto treated as an equality constraint. Özkaya and Gauger [106] later proposed a penalty multiplier method to treat this inequality constraint. Recently, the convergence theory of Hamdi and Griewank has been extended to equality-constrained optimization in ref. [147]. Extensions of this theory to inequality constraints have not been reported yet.

The divertor design problem is converted in this work into a PDE-constrained optimization problem with multiple nonlinear 'design' constraints. Since this problem therefore differs fundamentally from the problems treated with one-shot optimization so far, an outlook is given in this section on how to solve the problem in a one-shot fashion. More specifically, a practical approach is proposed here that reduces the nonlinear inequality-constrained optimization problem to solving a simple quadratic problem in each one-shot iteration. Although a general convergence theory for inequality constrained one-shot optimization is out of the scope of this thesis, it is explained how the line search globalization can be adapted to account for the constraint feasibility. Further elaboration and testing of the one-shot solution to the inequality constrained divertor design problem is left for future work.

7.6.1 Constrained one-shot optimization based on an active set strategy

In section 2.1.1, the KKT conditions of an inequality constrained optimization problem were listed. Given a discrete vector of constraints $\boldsymbol{h}(\boldsymbol{\varphi}) \leqslant 0$ on the design space, the Lagrangian of the constrained optimization problem reads

$$\mathcal{L}_c(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*, \boldsymbol{\mu}) = \mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}) + \langle \boldsymbol{q}^*, \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) \rangle + \boldsymbol{h}^\top(\boldsymbol{\varphi})\boldsymbol{\mu}.$$

In addition to the classical conditions

$$\begin{cases} \nabla_{\boldsymbol{q}^*}\mathcal{L}_c = \boldsymbol{c}\,(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}) = 0 \\ \nabla_{\boldsymbol{q}}\mathcal{L}_c = \nabla_{\boldsymbol{q}}\mathcal{J}(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}) + \partial_{\boldsymbol{q}}\boldsymbol{c}^*(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^* = 0 \\ \nabla_{\boldsymbol{\varphi}}\mathcal{L}_c = \nabla_{\boldsymbol{\varphi}}\mathcal{J}(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}) + \partial_{\boldsymbol{\varphi}}\boldsymbol{c}^*(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^* + \partial_{\boldsymbol{\varphi}}\boldsymbol{h}^\top(\bar{\boldsymbol{\varphi}})\bar{\boldsymbol{\mu}} = 0 \end{cases},$$

three conditions appear that are related to the inequality constraints:

$$\begin{cases} \boldsymbol{h}(\bar{\boldsymbol{\varphi}}) \leqslant 0 \\ \bar{\boldsymbol{\mu}} \geqslant 0 \\ \bar{\boldsymbol{\mu}}^\top \boldsymbol{h}(\bar{\boldsymbol{\varphi}}) = 0 \end{cases}.$$

As these last three KKT-conditions are non-smooth, they cannot be treated directly by an approximate Newton method. Therefore, a so-called *active set* method may be applied to divide the constraints in a set of active and inactive

constraints. If a component h_i of the vector $\boldsymbol{h}$ equals zero, then it is considered active. Conversely, if h_i is strictly smaller then zero, it is considered inactive. Formally, the index sets $\mathbb{A} \subset \{1, \dots, n_h\}$ and $\mathbb{I} = \{1, \dots, n_h\} \backslash \mathbb{A}$ can be introduced to denote the indices of te active and inactive constraints at the optimum. As such, the constraint vector $\boldsymbol{h} = (h_1, \dots, h_{n_h})^\top$ equals

$$\boldsymbol{h} = \begin{pmatrix} \boldsymbol{h}_\mathbb{A} \\ \boldsymbol{h}_\mathbb{I} \end{pmatrix}. \tag{7.53}$$

The question of interest is of course how to find this active set. Assuming for now that the sets $\mathbb{A}$ and $\mathbb{I}$ are known, the Lagrangian of the constrained optimization problem reduces to

$$\mathcal{L}_c(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*, \boldsymbol{\mu}_\mathbb{A}) = \mathcal{J}(\boldsymbol{\varphi}, \boldsymbol{q}) + \langle \boldsymbol{q}^*, \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) \rangle + \boldsymbol{h}_\mathbb{A}^\top(\boldsymbol{\varphi}) \boldsymbol{\mu}_\mathbb{A}.$$

A minimizer $(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}, \bar{\boldsymbol{q}}^*, \bar{\boldsymbol{\mu}})$ is then found if and only if there exist index sets $\mathbb{A}$ and $\mathbb{I}$ so that

$$\begin{cases} \nabla \mathcal{L}_c(\bar{\boldsymbol{\varphi}}, \bar{\boldsymbol{q}}, \bar{\boldsymbol{q}}^*, \boldsymbol{\mu}_\mathbb{A}) = 0 \\ \boldsymbol{h}_\mathbb{I}(\bar{\boldsymbol{\varphi}}) < 0 \\ \bar{\boldsymbol{\mu}}_\mathbb{A} > 0 \\ \bar{\boldsymbol{\mu}}_\mathbb{I} = 0 \end{cases}.$$

If these index sets are known, one may use an approximate rSQP approach to solve for $\nabla \mathcal{L}_c = 0$. This approximate rSQP matrix then reads

$$\begin{bmatrix} 0 & 0 & 0 & A^* \\ 0 & B & \partial_{\boldsymbol{\varphi}} \boldsymbol{h}_\mathbb{A}^\top & \partial_{\boldsymbol{\varphi}} \boldsymbol{c}^* \\ 0 & \partial_{\boldsymbol{\varphi}} \boldsymbol{h}_\mathbb{A} & 0 & 0 \\ A & \partial_{\boldsymbol{\varphi}} \boldsymbol{c} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{q} \\ \Delta \boldsymbol{\varphi} \\ \Delta \boldsymbol{\mu}_\mathbb{A} \\ \Delta \boldsymbol{q}^* \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{q}} \mathcal{L}_c \\ -\nabla_{\boldsymbol{\varphi}} \mathcal{L}_c \\ -\boldsymbol{h}_\mathbb{A} \\ -\boldsymbol{c} \end{bmatrix}, \tag{7.54}$$

with A and A^* being approximations of $\partial_{\boldsymbol{q}} \boldsymbol{c}$ and $\partial_{\boldsymbol{q}} \boldsymbol{c}^*$ that include relaxation and account for the iterative state and adjoint solver. The linearized model equations have been used here (cfr. section 2.2.2) to eliminate the dependence on $\boldsymbol{q}(\boldsymbol{\varphi})$ and obtain the exact reduced Hessian

$$\begin{aligned} B = Z^\top H Z = \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}} \mathcal{L}_c - \nabla_{\boldsymbol{\varphi}\boldsymbol{q}} \mathcal{L}_c \, \partial_{\boldsymbol{q}} \boldsymbol{c}^{-1} \partial_{\boldsymbol{\varphi}} \boldsymbol{c} - \left(\partial_{\boldsymbol{q}} \boldsymbol{c}^{-1} \partial_{\boldsymbol{\varphi}} \boldsymbol{c} \right)^\top \nabla_{\boldsymbol{q}\boldsymbol{\varphi}} \mathcal{L}_c + \\ \left(\partial_{\boldsymbol{q}} \boldsymbol{c}^{-1} \partial_{\boldsymbol{\varphi}} \boldsymbol{c} \right)^\top \nabla_{\boldsymbol{q}\boldsymbol{q}} \mathcal{L}_c \, \partial_{\boldsymbol{q}} \boldsymbol{c}^{-1} \partial_{\boldsymbol{\varphi}} \boldsymbol{c}, \end{aligned} \tag{7.55}$$

with

$$H := \begin{bmatrix} \nabla_{\boldsymbol{q}\boldsymbol{q}}\mathcal{L}_c & \nabla_{\boldsymbol{q}\boldsymbol{\varphi}}\mathcal{L}_c \\ \nabla_{\boldsymbol{\varphi}\boldsymbol{q}}\mathcal{L}_c & \nabla_{\boldsymbol{\varphi}\boldsymbol{\varphi}}\mathcal{L}_c \end{bmatrix} \tag{7.56}$$

and

$$Z := \begin{bmatrix} \partial_{\boldsymbol{\varphi}}\boldsymbol{q} \\ \mathbb{I} \end{bmatrix} = \begin{bmatrix} -\partial_{\boldsymbol{q}}\boldsymbol{c}^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c} \\ \mathbb{I} \end{bmatrix}. \tag{7.57}$$

Ref. [79] indicates, however, that for the approximate rSQP solver, fastest convergence is achieved using the approximate reduced Hessian

$$B_A = Z_A^\top H Z_A, \tag{7.58}$$

with

$$Z_A := \begin{bmatrix} -A^{-1}\partial_{\boldsymbol{\varphi}}\boldsymbol{c} \\ \mathbb{I} \end{bmatrix}. \tag{7.59}$$

Notice that, since the constraints $\boldsymbol{h}_{\mathbb{A}} = 0$ are not eliminated from the original problem, this method is in literature referred to as a *partially reduced SQP* (prSQP) method. The Hessian B_A can thus be estimated using a BFGS method like the one outlined in section 7.4.4, but now using the gradient of the constrained Lagrangian $\nabla_{\boldsymbol{\varphi}}\mathcal{L}_c$ in the BFGS formula.

A Gauss-Seidel iteration can be used to solve the approximate prSQP problem (7.54) in a one-shot manner. The first row then corresponds to a step of the adjoint solver. The second and third row have to be solved jointly, since the prSQP matrix is not lower triangular. After solving the adjoint equations for a new costate $\boldsymbol{q}^*_{k+1}$ these two rows read

$$\begin{bmatrix} B_k & \partial_{\boldsymbol{\varphi}}\boldsymbol{h}_{\mathbb{A}}^\top(\boldsymbol{\varphi}_k) \\ \partial_{\boldsymbol{\varphi}}\boldsymbol{h}_{\mathbb{A}}(\boldsymbol{\varphi}_k) & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\varphi} \\ \Delta\boldsymbol{\mu}_{\mathbb{A}} \end{bmatrix} = \begin{bmatrix} -\nabla_{\boldsymbol{\varphi}}\mathcal{L}_c(\boldsymbol{\varphi}_k, \boldsymbol{q}_k, \boldsymbol{q}^*_{k+1}, \boldsymbol{\mu}_k) \\ -\boldsymbol{h}_{\mathbb{A}}(\boldsymbol{\varphi}_k) \end{bmatrix}. \tag{7.60}$$

In the right hand side of this equation, we can formally rewrite the first component as

$$-\nabla_{\boldsymbol{\varphi}}\mathcal{L}_c = -\nabla_{\boldsymbol{\varphi}}\mathcal{L} - \partial_{\boldsymbol{\varphi}}\boldsymbol{h}^\top(\boldsymbol{\varphi})\bar{\boldsymbol{\mu}}, \tag{7.61}$$

where

$$\mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{q}, \boldsymbol{q}^*) = \mathcal{I}(\boldsymbol{\varphi}, \boldsymbol{q}) + \langle \boldsymbol{q}^*, \boldsymbol{c}(\boldsymbol{\varphi}, \boldsymbol{q}) \rangle$$

is the Lagrangian of the optimization problem excluding design constraints. Additionally, $\nabla_{\boldsymbol{\varphi}}\mathcal{L}$ approaches $\nabla\hat{\mathcal{I}}$ as state and adjoint converge. Therefore, these

two equations simply solve the quadratic optimization problem

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \nabla\hat{\mathfrak{J}}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T B_k \boldsymbol{p} \\ s.t. \quad & \boldsymbol{h}_{\mathbb{A}}(\boldsymbol{\varphi}_k) + \nabla\boldsymbol{h}_{\mathbb{A}}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} = 0, \end{aligned} \tag{7.62}$$

with $p = \Delta\boldsymbol{\varphi}$ the design step. Moreover, since Robinson [117] showed that the active set of this quadratic problem converges to the active set of the original nonlinear program, the second step of the one-shot algorithm comprises solving the quadratic problem

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \nabla\hat{\mathfrak{J}}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T B_k \boldsymbol{p} \\ s.t. \quad & \boldsymbol{h}(\boldsymbol{\varphi}_k) + \nabla\boldsymbol{h}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} \leqslant 0. \end{aligned} \tag{7.63}$$

Finally, solving the last row of the prSQP matrix corresponds to an iteration of the state equation at $\boldsymbol{\varphi}_{k+1}, \boldsymbol{q}_k, \boldsymbol{q}^*_{k+1}$.

Reordering thus leads to the following one-shot optimization algorithm

Algorithm 7.3. *Constrained one-shot algorithm*

1. *Make a pseudo-time step Δt in the state equations to update $\boldsymbol{q}$.*
2. *Make a pseudo-time step Δt in the adjoint equations to update $\boldsymbol{q}^*$.*
3. *Update/calculate the design preconditioner B*
4. *Solve the quadratic problem*

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \nabla\hat{\mathfrak{J}}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T B_k \boldsymbol{p} \\ s.t. \quad & \boldsymbol{h}(\boldsymbol{\varphi}_k) + \nabla\boldsymbol{h}\left(\boldsymbol{\varphi}_k\right)^T \boldsymbol{p} \leqslant 0. \end{aligned} \tag{7.64}$$

5. *update $\boldsymbol{\varphi}$.*

It is clear that this algorithm strongly resembles the nested inequality-constrained optimization algorithm outlined in section 4.5. In fact, the algorithm can be kept exactly the same but just requires reducing the number of state and adjoint iterations to a single iteration per optimization step. Notice that if constraints are imposed that depend on the state variables $\boldsymbol{q}$, the algorithm is to

be augmented with an additional adjoint iteration for each state constraint between steps 2 and 3 in the above algorithm. These additional adjoint equations of the constraints then need to be converged along with state, adjoint, and design equation in order to converge to a solution of the KKT conditions.

7.6.2 Globalization of the constrained one-shot method

In the spirit of the globalized one-shot of section 7.4.2, one may consider adding a line search method based on the augmented Lagrangian (7.51). To cope with the additional constraints in line searches, several options exist. One might further augment this augmented Lagrangian merit function with a penalty term for feasibility of the design constraints. The reader is referred to Ref. [103] for an overview of penalty functions. Such a term is in general non-smooth for inequality constraints. Therefore, one might consider using a slack variable approach to translate them into equality constraints during line searches. Alternatively, one might resort to a projection method. However, solving the full nonlinear projection problem (4.7) might be computationally very demanding in this one-shot optimization setting. Nevertheless, this might be compensated by the smaller design steps (and thus relatively low infeasibility and low number of required projection iterations) that are taken in one-shot optimization and the still relatively cheap constraint evaluations that do neither require plasma edge grid evaluation nor evaluations of the plasma solver.

7.7 Discussion and conclusions

In this chapter, one-shot methods are considered for use in magnetic divertor design. Throughout the chapter, both the reduction in computational cost compared to nested optimization methods and the robustness of one-shot convergence are central considerations. First tests with a relatively simple method based on the method of Hazra et al. [69] indicated that the potential speed gain of one-shot methods is significant, but that slow grid generation steps impede reaching its full potential. Therefore, a grid deformation method has been elaborated for flux aligned grids. This reduced the wall-clock time needed to make a grid by a

factor of 10. Thus, both cost function and sensitivity calculations are significantly accelerated.

Further acceleration might be reached using multi-step one-shot methods. Using such methods, it can be entirely avoided that the costs of the design step dominates the one-shot method. One could for example choose the number of state and adjoint iterations performed next to one design update to spend an equal amount of time on state, adjoint, and design updates. With the grid deformation method at hand, also a full adjoint sensitivity calculation comes within reach. Both multi-step one-shot methods and full adjoint sensitivity calculations are interesting topics for further research.

Looking at computational cost solely, the one-shot method presents itself at present already as an attractive alternative for nested optimization methods. From a convergence point of view, however, robust convergence is difficult to achieve in magnetic divertor design. Discontinuities in design gradients have been found to be the source of convergence irregularities. Although the presence of these discontinuities could be significantly reduced using a plasma edge grid extending up to the walls, removing all sources of discontinuities might be a very demanding task. Nevertheless, for well-chosen parameters, the one-shot method of Hazra et al. converges to the same design variables as the nested optimization studies. It may, however, be questioned whether it is not rather the choice of test case that tolerates the very noisy convergence behaviour. For a test case with more complicated physics, such as a detached test case, the contribution of cost function changes due to state convergence is expected to increase. As such, the irregular state convergence observed using the one-shot method of Hazra et al. could affect the overall one-shot convergence. Since such detached cases are computationally very demanding, they are left for future research.

In an attempt to achieve fast and guaranteed convergence, a novel globalized one-shot method has been developed. In comparison to state-of-the-art globalized one-shot methods, it avoids the need of second order adjoint derivatives and reduces line search costs. A secant rule for one-shot BFGS updates is then consistently derived that avoids multiple sensitivity calculations per one-shot iteration. Techniques from damped BFGS updates are adapted to make the Hessian estimation bounded and more robust. The method is subsequently combined with

practical in-parts adjoint sensitivity calculations of the augmented Lagrangian. To obtain good accuracy for the finite difference part of these sensitivity calculations, a semi-analytical approach is applied. In presence of gradient discontinuities, the globalization strategy achieves robust convergence of state and adjoint equations. This is a clear asset of the novel one-shot method. Nevertheless, local optima seem to impede convergence to the same optimum as those of the nested optimization if high state feasibility penalties are used. In conclusion, the method holds promising results for use in more test cases or applications that encounter difficulties achieving convergence. Yet, for magnetic divertor design, a further examination should aim to find the sources of the local optima, and to eliminate them if possible.

Finally, an outlook has been given on an approach to embed additional design constraints in the one-shot procedure. The proposed method would in principle be able to completely reuse the inequality constrained approach of the nested optimization algorithm. If desired, the method could be combined with the novel globalization strategy, though convergence theory for inequality-constrained one-shot optimization is to be further developed. Alternatively, further testing is needed to assess the convergence of the constrained one-shot approach.

8

Discussion and Conclusions

8.1 General conclusions

One of the key challenges on the road to commercial fusion power plants is finding a solution for the divertor power and particle exhaust. Therefore, improved magnetic configurations are currently high-priority research in the fusion community. Motivated by the recent successful formulation of divertor structure design as a shape optimization problem, this dissertation examined the applicability of optimal design methods to magnetic divertor design. One of the main challenges in this application is the immense computational cost associated to the complex multi-fluid codes that describe the transport processes in the plasma edge. Given the cost-efficiency of adjoint-based gradient calculation, this approach is pursued in this dissertation to achieve a practical optimization method.

Because of the novelty in the nuclear fusion community, the main concepts of optimization with partial differential equations were carefully introduced. The remainder of the dissertation described the development of models and techniques for the first automated magnetic divertor design strategy. In this section, general conclusions are first formulated with respect to modelling and divertor design. Then, a second part focuses on the more technical conclusions in regard to development of optimization algorithms.

Modelling and magnetic divertor designs

The first step comprises the development of an integrated modelling approach for magnetic equilibrium and plasma edge transport. An intuitive perturbation model for magnetic field calculations is combined with a reduced plasma edge transport model in first magnetic design trials. Moreover, due to the strong flow anisotropy in fusion tokamaks, the plasma edge transport calculations are preferentially discretized on a curvilinear grid that is aligned to the poloidal magnetic flux. The grid generator as such becomes an important link in the integrated model. A unique plasma edge grid generator is therefore developed that complies with the requirement for an automated model. This integrated model is the first tool that automatically allows the assessment of external coil changes on the divertor heat exhaust.

Subsequently, the magnetic field model is revisited. A comparison is made between the perturbation model and a free boundary equilibrium model. After analyzing the different magnetic fields and their sensitivities, it is concluded that a model for the core plasma currents is indispensable for accurate magnetic flux calculations in the plasma edge. A MATLAB version of the CEDRES++ finite element code for FBE calculations is therefore integrated into the model. To achieve a fully automated coupling with the current-free plasma edge transport calculations, an adaptive mesh refinement strategy is implemented in the magnetic equilibrium code in collaboration with Holger Heumann (INRIA Sophia-Antipolis, France). As such, accurate discretization of the plasma edge transport equations is also obtained in the regions around the X-points.

Based on these models, first optimal design studies are performed. A tracking-type cost function is defined that aims for uniform heat loads onto the target surface. To obtain physically relevant results, appropriate penalty terms are included that avoid significant parallel heat loads to the sensitive first wall components outside the divertor targets as well as excessive Joule losses in the divertor coils. By imposing constraints, it is guaranteed that the thermal limits of the coils are respected and that a single-null divertor configuration is kept. Additional constraints are imposed to account for the model validity.

Realistic test cases are set up in this dissertation for design of the JET and WEST magnetic divertors. These first trials demonstrate the potential of the proposed schemes for magnetic divertor design. Plausible design trends are found that strongly resemble the X-divertor concept. Since these results, however, depend on the plasma parameters, this should not be considered as a general conclusion. Both the perturbation model and the FBE model show this trend, but the currents needed to realize such configurations are seriously overestimated using the perturbation model. For this reason, the author recommends including an FBE code for steady-state magnetic divertor design. It is clear that this augments the code complexity. On the other hand, such FBE calculations are very fast and do therefore barely contribute to the computational cost.

Although in this thesis X-divertor-like concepts are found, less intuitive divertor concepts might be retrieved in a plasma regime close to detachment. It should be noted that the presented test cases (JET and WEST) as well as the code implementation exclude interesting snowflake-type magnetic configurations. Yet, a peak heat load reduction of over 50% is found for the WEST case, indicating a great potential of these methods to assist divertor design. Furthermore, it is illustrated how the choice of penalty factors leads to different optimal target profiles that are both optimal, depending on whether the method is applied to an existing or new reactor. Eventually, a true multi-objective optimization approach might be considered.

Development of optimal design algorithms

Given the complex and CPU-demanding nature of plasma edge transport simulations, robust, efficient, and implementation-friendly algorithms are pursued to enable computational magnetic divertor design.

First, a nested quasi-Newton type SQP optimization algorithm is implemented to deal with the nonlinearly-constrained optimization problem. In a first stage, sensitivities of the integrated code are calculated automatically using finite difference calculations. To ensure robustness, an efficient line search algorithm by Moré and Thuente [99] is adapted with a nonlinear gradient projection method

and generalized Wolfe conditions. As such, constraint feasibility can be guaranteed at every instance in the optimization algorithm. This optimization strategy is found to achieve robust convergence to the constrained optimum.

Next, the efficiency is significantly increased by replacing the finite difference gradients with adjoint-based gradient calculations. However, moving domain boundaries and hidden source functions hinder a full adjoint sensitivity calculation. Therefore, a practical in-parts adjoint sensitivity calculation is elaborated. This leads to a semi-discrete approach, in which the existing continuous adjoint plasma edge solver can be used in combination with practical discrete finite difference calculations for the remaining sensitivity contribution. It is shown in this dissertation that the consistent treatment of the boundary terms in the sensitivity calculation needs special care. This is guaranteed in practice by a slight modification to the boundary condition implementation. Asymptotic grid convergence is demonstrated between in-parts adjoint and full finite difference sensitivity calculations. Because of the relatively low computational cost of grid generator and magnetic field evaluations, nearly the same efficiency gain is achieved as compared to full adjoint sensitivity calculations. Furthermore, in contrast to standard adjoint methods, the strategy circumvents analytical derivations of the design sensitivity expression. As such, after evaluation of the adjoint plasma state, a virtually unlimited set of parameter sensitivities can be calculated without any derivational effort and at low computational cost.

Finally, one-shot methods are studied for optimal magnetic divertor design. Using an unconstrained design case, a strong reduction of the number of system iterations is already found for a standard one-shot approach. Despite, gradient discontinuities in the grid generator software are found to disturb convergence. Moreover, the increased number of mesh generation events added to the computational cost of the one-shot algorithm. A further contribution of this thesis therefore focusses on exploiting the full potential of one-shot optimization by reducing the computational cost of design steps. To this end, a grid deformation strategy is developed for flux-aligned curvilinear grids. The strict necessity for alignment is efficiently combined with smooth deformation and solved with a Gauss-Seidel type iterative solver. The wall-clock time for constructing new grids is as such reduced by an order of magnitude.

In comparison to a nested optimization method, the one-shot method was found to reduce computational costs. Despite, it sacrifices the monotonic convergence that the nested method attains using globalization strategies. For the cases studied, the one-shot strategy still tends to converge to the same optimal configurations as found by the nested algorithm. This, however, might no longer be the case when optimizing under detached plasma conditions, since these conditions yield higher dependence of the heat load on the collisional processes and thus state convergence. A novel one-shot method has therefore been derived that combines state-of-the-art globalization techniques for one-shot methods with increased efficiency and a practical implementation. The method searches for descent on an augmented Lagrangian function that includes a penalty based on the normalized residuals of the transport equations. In comparison to state-of-the-art one-shot methods, it avoids the implementation and calculation of second order adjoint contributions. Furthermore, a consistent BFGS approach is derived for superlinear convergence in one-shot optimization along with practical Hessian bounds that increase the robustness. A simple and inexpensive line search algorithm provides globalization. The method is applied to the unconstrained design case using an accurate semi-analytical version of the in-parts adjoint sensitivity calculations. Robust convergence is indeed achieved. Yet, it is found that high penalties and thus small design steps increase the chance to get stuck in a local optimum, probably created by the gradient noise. Further steps might therefore be needed for application to magnetic divertor design. Nevertheless, the method shows great assets for design applications with difficult one-shot convergence behaviour.

8.2 Suggestions for further research

In this dissertation, the focus was on developing methods for computational "optimal magnetic divertor design" and examining their potential. Since promising results are achieved, further accuracy of the designs might now be aimed at by using a state-of-the-art plasma model. Some model extensions, like modelling of impurities and kinetic neutrals might strongly influence the target heat load since they might trigger the detachment regime. One possibility would be to involve the full B2-EIRENE code in the design cycle. However, to enable these adjoint

optimization methods for the B2-EIRENE code [115], further research is still needed. First of all, the correct adjoint complement of such a coupled finite volume method/Monte-Carlo simulation should be derived. Secondly, the effect of statistical noise on the design gradient needs to be studied. Ease of maintenance will furthermore be of great importance given the size and worldwide usage of this code. In this regard, it would be already of great advantage if the boundary conditions could be set automatically based on the choice of forward boundary conditions. In principle, the general adjoint boundary condition formulation derived in chapter 5 could be used for this purpose. It remains to be verified though whether this also holds for Dirichlet conditions, which are imposed in a numerical way using very big fluxes (see section 3.3.4). In regard to ease of maintenance, also backward mode AD could be considered as an alternative to the continuous adjoint method [63]. However, due to the memory-intensity of these approaches, one might rather further develop a hybrid approach [137]. In DEMO, neutral interactions will become significant such that a fluid assumption might not be such a bad approximation after all. Therefore, extending the neutral model with a momentum equation and energy equation such as in Ref. [76] could be done as an intermediate step with much lower efforts. Finally, the optimal design method would benefit from a *wide grid* approach, extending the plasma edge simulation up to the vessel walls [4].

Furthermore, one of the final goals should be assessing and improving novel magnetic configurations, such as snowflake divertors. At present, such plasma edge simulations are out of the scope of the B2-EIRENE code (and its reduced MATLAB version). The snowflake divertor poses challenges for the cut approach that lies at the basis of the curvilinear grid structure. Extensions to a more general treatment have been initiated by Van den Kerkhof [144]. However, the presence of the second-order null in poloidal magnetic field might also require revisiting the modelling of radial transport using a spatially constant anomalous diffusion coefficient. Indeed, experiments indicate a strong increase of radial transport due to the low poloidal field at the snowflake X-point [95].

Up to now, the cost function is chosen to solely reflect the heat exhaust design criterion. Nevertheless, other design goals might equally well be incorporated

in the optimal design framework as alternative design goals, penalty terms, design constraints, or possibly even in a multi-objective optimization approach. Optimizing the pumping capabilities of Helium ashes is one of the fascinating possibilities of these design methods. Studies to enable adjoint optimization of the Helium exhaust have been initiated by Dedoncker [36]. Another example is the stability of the plasma operation, which imposes constraints on the magnetic design. For example, designs sensitive to MHD instabilities could be excluded from the optimization by extending the model with an MHD stability code (see e.g. [9, 77]). Approximate relations for simplified geometries such as in Ref. [148, 149] might provide an alternative. Furthermore, a threshold on the vertical plasma acceleration could be included to account for vertical stability. In controllers, they are generally substituted by constraints on the plasma shape, the so-called *plasma elongation* [35, 132].

The practical in-parts adjoint approach also makes some other interesting applications possible. It is indeed a practical way to calculate several sensitivities to virtually any parameter without the derivational effort of a regular continuous adjoint. As such, it might serve well for robust design with respect to uncertain model parameters, e.g. anomalous transport coefficients or core boundary conditions. These robust design methods aim at achieving optimal configurations with lower sensitivity to small variations of these parameters. Robust design studies for target shape optimization have been initiated by Horsten and Bauweraerts [75]. Similarly, in-parts adjoint sensitivities could be exploited for parametric inference of model coefficients from experiments. Such parametric inference methods might be of primordial importance to analyse experiments, by calibrating transport coefficients or even build correlations. First steps towards an adjoint-based estimation of anomalous transport coefficients and a number of boundary conditions have been taken in [5, 45, 47].

Finally, ever more efficient algorithms should be pursued for plasma edge simulation and optimization. Indeed, B2-EIRENE simulations of the DEMO reactor come at a tremendous computational cost due to the high neutral collisionality. With respect to the optimization methods for magnetic divertor design, the investigation of one-shot methods for optimization with design constraints could be continued. Especially multi-step one-shot algorithms are expected to yield

increased efficiency. Similarly, full adjoint strategies for sensitivity calculations based on the novel grid deformation method would give a further acceleration of the one-shot algorithm. Of course, the primary source of the high computational cost remains the plasma edge code itself. Therefore, research should primarily aim at speeding up this code. Several options are currently under investigation as a part of the European fusion work plan [50, 56].

Bibliography

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

[2] Artaud, J., Basiuk, V., Imbeaux, F., Schneider, M., Garcia, J. *et al.* (2010). The CRONOS suite of codes for integrated tokamak modelling. *Nuclear Fusion*, **50**, 043001.

[3] Baelmans, M. (1993). *Code Improvements and Applications of a Two-dimensional Edge Plasma Model for Toroidal Devices*. Ph.D. thesis, Departement Mechanica, Faculteit der Toegepaste Wetenschappen, Katholieke Universiteit Leuven.

[4] Baelmans, M., Börner, P., Dekeyser, W. & Reiter, D. (2011). Tokamak plasma edge modelling including the main chamber wall. *Nuclear Fusion*, **51**, 083023.

[5] Baelmans, M., Blommaert, M., De Schutter, J., Dekeyser, W. & Reiter, D. (2014). Efficient parameter estimation in 2D transport models based on an adjoint formalism. *Plasma Physics and Controlled Fusion*, **56**, 114009.

[6] Balescu, R. (1988). *Transport processes in plasmas*. North-Holland, Netherlands.

[7] Bänsch, E. (1991). Local mesh refinement in 2 and 3 dimensions. *Impact Comput. Sci. Engrg.*, **3**, 181–191.

[8] Batina, J.T. (1990). Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, **28**.

[9] Blokland, J., van der Holst, B., Keppens, R. & Goedbloed, J. (2007). PHOENIX: MHD spectral code for rotating laboratory and gravitating astrophysical plasmas. *Journal of Computational Physics*, **226**, 509–533.

[10] Blom, F.J. (2000). Considerations on the spring analogy. *Int. J. Numer. Meth. Fluids*, **32**, 647–668.

[11] Blommaert, M. (2012). *Optimization of the magnetic divertor configuration for nuclear fusion reactors*. Master's thesis, Department of mechanical engineering, KU Leuven.

[12] Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N. & Reiter, D. (2014). An Automated Approach to Magnetic Divertor Configuration Design. *Nuclear Fusion*, **55**.

[13] Blommaert, M., Baelmans, M., Dekeyser, W., Gauger, N. & Reiter, D. (2015). A novel approach to magnetic divertor configuration design. *Journal of Nuclear Materials*, **463**, 1220–1224.

[14] Blommaert, M., Baelmans, M., Heumann, H., Marandet, Y., Bufferand, H. *et al.* (2016). Magnetic Field Models and their Application in Optimal Magnetic Divertor Design. *Contributions to Plasma Physics*, **56**, 796–801.

[15] Blommaert, M., Heumann, H., Baelmans, M., Gauger, N.R. & Reiter, D. (2016). Towards Automated Magnetic Divertor Design for Optimal Heat Exhaust. *ESAIM: Proceedings and Surveys*, **53**, 49–63.

[16] Blommaert, M., Reiter, D. & Baelmans, M. (2016). An Efficient Methodology to Analyze Plasma Edge Model Parameter Sensitivities. *Nuclear Materials and Energy*, in press.

[17] Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N.R. & Reiter, D. (2017). A practical globalization of one-shot optimization for optimal design of tokamak divertors. *Journal of Computational Physics*, **328**, 399–412.

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

[19] Blum, J., Le Foll, J. & Thooris, B. (1981). The self-consistent equilibrium and diffusion code SCED. *Computer Physics Communications*, **24**, 235 – 254.

[20] Blum, J., Boulbe, C. & Faugeras, B. (2012). Reconstruction of the equilibrium of the plasma in a Tokamak and identification of the current density profile in real time . *Journal of Computational Physics*, **231**, 960 – 980.

[21] Borzì, V., A. & Schulz (2011). *Computational Optimization of Systems Governed by Partial Differential Equations*. Society for Industrial and Applied Mathematics.

[22] Bosse, T., Gauger, N.R., Griewank, A., Günther, S., Kaland, L. *et al.* (2014). Optimal Design with Bounded Retardation for Problems with Non-separable Adjoints.

[23] Bosse, T., Lehmann, L. & Griewank, A. (2014). Adaptive sequencing of primal, dual, and design steps in simulation based optimization. *Computational Optimization and Applications*, **57**, 731–760.

[24] Bourdelle, C., Artaud, J., Basiuk, V., Bécoulet, M., Brémond, S., *et al.* (2015). WEST Physics Basis. *Nuclear Fusion*, **55**, 063017.

[25] Braginskii, S.I. (1965). Transport Processes in a Plasma. *Reviews of Plasma Physics*, **1**, 205.

[26] Bucalossi, J., Missirlian, M., Moreau, P., Samaille, F., Tsitrone, E. *et al.* (2014). The WEST project: Testing ITER divertor high heat flux component technology in a steady state tokamak environment. *Fusion Engineering and Design*, **89**, 907–912.

[27] Bufferand, H., Bensiali, B., Bucalossi, J., Ciraolo, G., Genesio, P. *et al.* (2013). Near wall plasma simulation using penalization technique with the transport code SolEdge2D-Eirene. *Journal of Nuclear Materials*, **438, Supplement**, S445–S448.

[28] Bufferand, H., Bucalossi, J., Ciraolo, G., Fedorczak, N., Ghendrih, P. *et al.* (2014). Density Regimes and Heat Flux Deposition in the WEST Shallow Divertor Configuration. *Contrib. Plasma Phys.*, **54**, 378–382.

[29] Chankin, A., Coster, D., Dux, R., Fuchs, C., Haas, G. *et al.* (2006). SOLPS modelling of ASDEX upgrade H-mode plasma. *Plasma physics and controlled fusion*, **48**, 839.

[30] Chen, G. & Zhou, J. (1992). *Boundary element methods*. Academic Press, Ltd., London.

[31] Christianson, B. (1994). Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, **3**, 311–326.

[32] Coster, D., Basiuk, V., Pereverzev, G., Kalupin, D., Zagórksi, R. *et al.* (2010). The European Transport Solver. *IEEE Transactions on Plasma Science*, **38**, 2085–2092.

[33] Coster, D.P., Bonnin, X., Braams, B., Reiter, D., Schneider, R. & the ASDEX Upgrade Team (2004). Simulation of the Edge Plasma in Tokamaks. *Physica Scripta*, **2004**, 7–13.

[34] Cowley, S. (2010). Hot fusion. *Physics World*, **23**, 46–51.

[35] Crisanti, F., Albanese, R., Ambrosino, G., Ariola, M., Lister, J. *et al.* (2003). Upgrade of the present JET shape and vertical stability controller. *Fusion engineering and design*, **66**, 803–807.

[36] Dedoncker, S. (2016). *Towards Optimization of the Helium Exhaust from Nuclear Fusion Divertors*. Master's thesis, Department of mechanical engineering, KU Leuven, in preparation.

[37] Dekeyser, W., Reiter, D. & Baelmans, M. (2012). Divertor Design through Shape Optimization. *Contrib. Plasma Phys.*, **52**, 544–549.

[38] Dekeyser, W., Reiter, D. & Baelmans, M. (2014). Automated divertor target design by adjoint shape sensitivity analysis and a one-shot method. *Journal of Computational Physics*, **278**, 117–132.

[39] Dekeyser, W., Reiter, D. & Baelmans, M. (2014). Divertor target shape optimization in realistic edge plasma geometry. *Nuclear Fusion*, **54**, 073022.

[40] Dekeyser, W., Reiter, D. & Baelmans, M. (2014). A one shot method for divertor target shape optimization. *PAMM*, **14**, 1017–1022.

[41] Dekeyser, W., Reiter, D. & Baelmans, M. (2014). *Optimal Plasma Edge Configurations for Next-Step Fusion Reactors (Optimale plasmarand-configuraties voor nieuwe generatie fusiereactoren)*. Ph.D. thesis, KU Leuven.

[42] Dekeyser, W., Reiter, D. & Baelmans, M. (2014). Optimal shape design for divertors. *Int. J. Computational Science and Engineering*, **9**, 397–407.

[43] Dekeyser, W., Reiter, D. & Baelmans, M. (2015). Designing divertor targets for uniform power load. *Journal of Nuclear Materials*, **463**, 1243–1247.

[44] Delport, S., Meyers, J. & Baelmans, M. (2010). *Optimal Control of a Turbulent Mixing Layer (Optimale benvloeding van een turbulente menglaag)*. Ph.D. thesis, KU Leuven.

[45] Deschutter, J. (2013). *Sensitivity analysis of optimal divertor configurations in nuclear fusion reactors*. Master's thesis, Department of mechanical engineering, KU Leuven.

[46] D'haeseleer, W., Hitchon, W., Callen, J. & Shohet, J. (1991). *Flux Coordinates and Magnetic Field Structure*. Springer-Verlag, Berlin.

[47] Dilissen, R. (2015). *Parameter estimation for calibration of plasma edge codes*. Master's thesis, Department of Mechanical Engineering, KU Leuven.

[48] Dnestrovskii, Y.N. & Kostomarov, D.P. (1986). Controlled Fusion and Numerical Simulation. In *Springer Series in Computational Physics*, 1–28, Springer Berlin Heidelberg.

[49] Dwight, R.P. (2009). Robust Mesh Deformation using the Linear Elasticity Equations. In H. Deconinck & E. Dick, eds., *Computational Fluid Dynamics 2006*, 401–406, Springer Berlin Heidelberg.

[50] EUROFUSION (2013). Work plan for the implementation of the fusion roadmap in 2014-2018.

[51] Federici, G., Kemp, R., Ward, D., Bachmann, C., Franke, T. *et al.* (2014). Overview of EU DEMO design and r&d activities. *Fusion Engineering and Design*, **89**, 882–889.

[52] Frerichs, H., Reiter, D., Schmitz, O., Evans, T. & Feng, Y. (2010). Three-dimensional edge transport simulations for DIII-D plasmas with resonant magnetic perturbations. *Nuclear Fusion*, **50**, 034004–.

[53] Fundamenski, W. (2010). *Power exhaust in fusion plasmas*. Cambridge University Press.

[54] Gauger, N.R., Griewank, A. & Riehme, J. (2008). Extension of fixed point PDE solvers for optimal design by one-shot method: With first applications to aerodynamic shape optimization. *European Journal of Computational Mechanics/Revue Europenne de Mcanique Numrique*, **17**, 87–102.

[55] Gherman, I. (2007). *Approximate Partially reduced SQP approaches of Aerodynamic Shape Optimization problems*. Ph.D. thesis, University of Trier, Faculty of Mathematics.

[56] Ghoos, K., Dekeyser, W., Samaey, G., Börner, P. & Baelmans, M. (2016). Accuracy and convergence of coupled finite-volume/Monte Carlo codes for plasma edge simulations of nuclear fusion reactors. *Journal of Computational Physics*, **322**, 162–182.

[57] Giannakoglou, K.C. & Papadimitriou, D.I. (2008). Adjoint Methods for Shape Optimization. In D. Thévenin & G. Janiga, eds., *Optimization and Computational Fluid Dynamics*, 79–108, Springer Berlin Heidelberg.

[58] Giles, M. & Pierce, N. (1997). Adjoint equations in CFD - Duality, boundary conditions and solution behaviour. In *Fluid Dynamics and Co-located Conferences*, American Institute of Aeronautics and Astronautics.

[59] Giles, M.B. & Pierce, N.A. (2000). An Introduction to the Adjoint Approach to Design. *Flow, Turbulence and Combustion*, **65**, 393–415.

[60] Grad, H. & Rubin, H. (1958). Hydromagnetic Equilibria and Force-Free Fields. *Proceedings of the 2nd UN Conf. on the Peaceful Uses of Atomic Energy*, **31**, 190.

[61] Grandgirard, V. (1999). *Modélisation de l'équilibre d'un plasma de tokamak.*. Ph.D. thesis, l'Université de Franche-Comté.

[62] Griewank, A. (2006). Projected Hessians for Preconditioning in One-Step One-Shot Design Optimization. In G. Di Pillo & M. Roma, eds., *Nonconvex Optimization and Its Applications*, vol. 83, 151–171, Springer US.

[63] Griewank, A. (2014). *Automatic Differentiation*. Princeton Companion to Applied Mathematics, Nicolas Higham Ed., Princeton University Press.

[64] Griewank, A. & Faure, C. (2002). Reduced Functions, Gradients and Hessians from Fixed-Point Iterations for State Equations. *Numerical Algorithms*, **30**, 113–139.

[65] Hamdi, A. & Griewank, A. (2010). Properties of an Augmented Lagrangian for Design Optimization. *Optimization Methods Software*, **25**, 645–664.

[66] Hamdi, A. & Griewank, A. (2011). Reduced quasi-Newton method for simultaneous design and optimization. *Computational Optimization and Applications*, **49**, 521–548.

[67] Haslinger, J. & Mäkinen, R. (2003). *Introduction to Shape Optimization*. Society for Industrial and Applied Mathematics.

[68] Hazra, S. & Schulz, V. (2004). Simultaneous Pseudo-Timestepping for PDE-Model Based Optimization Problems. *BIT Numerical Mathematics*, **44**, 457–472.

[69] Hazra, S., Schulz, V., Brezillon, J. & Gauger, N. (2005). Aerodynamic shape optimization using simultaneous pseudo-timestepping. *Journal of Computational Physics*, **204**, 46–64.

[70] Hazra, S.B. & Jameson, A. (2012). One-shot pseudo-time method for aerodynamic shape optimization using the Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, **68**, 564–581.

[71] Hertout, P., Boulbe, C., Nardon, E., Blum, J., Brémond, S. *et al.* (2011). The CEDRES++ equilibrium code and its application to ITER, JT-60SA and Tore Supra. *Fusion Engineering and Design*, **86**, 1045–1048.

[72] Heumann, H., Blum, J., Boulbe, C., Faugeras, B., Selig, G. *et al.* (2015). Quasi-static free-boundary equilibrium of toroidal plasma with CEDRES++: Computational methods and applications. *Journal of Plasma Physics*, **81**, 1469–7807.

[73] Hinze, M., Pinnau, R., Ulbrich, M. & Ulbrich, S. (2009). *Optimization with PDE constraints*, vol. 23 of *Mathematical Modelling: Theory and Applications*. Springer, New York.

[74] Hofmann, F. & Tonetti, G. (1988). Tokamak equilibrium reconstruction using Faraday rotation measurements. *Nuclear Fusion*, **28**, 1871.

[75] Horsten, n. & Bauweraerts, P. (2014). *Geometry and Development of a Navier-Stokes Fluid Neutral Model for Nuclear Fusion Reactors*. Master's thesis, Department of mechanical engineering, KU Leuven, in preparation.

[76] Horsten, N., Dekeyser, W., Samaey, G. & Börner, P. (2016). Fluid neutral model for use in hybrid simulations of a detached case. *Contributions to Plasma Physics*, in press.

[77] Huysmans, G.T.A., Sharapov, S.E., Mikhailovskii, A.B. & Kerner, W. (2001). Modeling of diamagnetic stabilization of ideal magnetohydrodynamic instabilities associated with the transport barrier. *Physics of Plasmas (1994-present)*, **8**, 4292–4305.

[78] Iollo, A., Kuruvila, G. & Ta'asan, S. (1995). Pseudo-Time Method for Optimal Shape Design Using the Euler Equations. Tech. rep., Institute for Computer Applications in Science and Engineering (ICASE).

[79] Ito, K., Kunisch, K., Schulz, V. & Gherman, I. (2010). Approximate Nullspace Iterations for KKT Systems. *SIAM. J. Matrix Anal. & Appl.*, **31**, 1835–1847.

[80] Jameson, A., Martinelli, L. & Pierce, N. (1998). Optimum Aerodynamic Design Using the Navier-Stokes Equations. *Theoretical and Computational Fluid Dynamics*, **10**, 213–237.

[81] Kaland, L. & Gauger, N. (2013). *The One-Shot Method: Function Space Analysis and Algorithmic Extension by Adaptivity*. Ph.D. thesis, RWTH Aachen University, Faculty of Mathematics, Computer Science and Natural Sciences.

[82] Kaland, L., De Los Reyes, J.C. & Gauger, N.R. (2013). One-shot methods in function space for PDE-constrained optimal control problems. *Optimization Methods and Software*, **29**, 376–405.

[83] Khayrutdinov, R. & Lukash, V. (1993). Studies of Plasma Equilibrium and Transport in a Tokamak Fusion Device with the Inverse-Variable Technique. *Journal of Computational Physics*, **109**, 193–201.

[84] Kotov, V., Reiter, D., Pitts, R., Jachmich, S., Huber, A. *et al.* (2008). Numerical modelling of high density JET divertor plasma with the SOLPS4. 2 (B2-EIRENE) code. *Plasma physics and controlled fusion*, **50**, 105012.

[85] Kotschenreuther, M., Valanju, P.M., Mahajan, S.M. & Wiley, J.C. (2007). On heat loading, novel divertors, and fusion reactors. *Physics of Plasmas*, **14**, 072502.

[86] Kukushkin, A., Pacher, H., Janeschitz, G., Coster, D., Reiter, D. & Schneider, R. (1999). Divertor performance in reduced-technical-objective/reduced-cost ITER. *Power*, **12**, 14.

[87] Kukushkin, A., Pacher, H., Kotov, V., Reiter, D., Coster, D. & Pacher, G. (2007). Effect of the dome on divertor performance in ITER. *Journal of Nuclear Materials*, **363-365**, 308–313.

[88] Kukushkin, A., Pacher, H., Loarte, A., Komarov, V., Kotov, V. *et al.* (2009). Analysis of performance of the optimized divertor in ITER. *Nuclear fusion*, **49**, 075008.

[89] Kukushkin, A., Pacher, H., Kotov, V., Pacher, G. & Reiter, D. (2011). Finalizing the ITER divertor design: The key role of SOLPS modeling. *Fusion Engineering and Design*, **86**, 2865–2873.

[90] Kuruvila, G., Ta'asan, S. & Salas, M. (1995). Airfoil design and optimization by the one-shot method. In *Aerospace Sciences Meetings*, American Institute of Aeronautics and Astronautics.

[91] Lao, L., Ferron, J., Groebner, R., Howl, W., John, H. *et al.* (1990). Equilibrium analysis of current profiles in Tokamaks. *Nuclear Fusion*, **30**, 1035.

[92] Lions, J. (1971). *Optimal control of systems governed by partial differential equations*. Springer-Verlag.

[93] Lipschultz, B., LaBombard, B., Terry, J., Boswell, C. & Hutchinson, I. (2007). Divertor physics research on Alcator C-Mod. *Fusion science and technology*, **51**, 369–389.

[94] Lunt, T., Feng, Y., Bernert, M., Herrmann, A., de Marné, P. *et al.* (2012). First EMC3-Eirene simulations of the impact of the edge magnetic perturbations at ASDEX Upgrade compared with the experiment. *Nuclear Fusion*, **52**, 054013.

[95] Lunt, T., Canal, G.P., Feng, Y., Reimerdes, H., Duval, B.P. *et al.* (2014). First EMC3-Eirene simulations of the TCV snowflake divertor. *Plasma Physics and Controlled Fusion*, **56**, 035009.

[96] Luxon, J. & Brown, B. (1982). Magnetic analysis of non-circular cross-section tokamaks. *Nuclear Fusion*, **22**, 813.

[97] Marchand, R. & Dumberry, M. (1996). CARRE: a quasi-orthogonal mesh generator for 2D edge plasma modelling. *Computer Physics Communications*, **96**, 232–246.

[98] Mc Carthy, P., Martin, P. & Schneider, W. (1999). The CLISTE Interpretive Equilibrium Code. Tech. Rep. IPP Report 5/85, Max-Planck-Institut fur Plasmaphysik.

[99] Moré, J.J. & Thuente, D.J. (1994). Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Softw.*, **20**, 286–307.

[100] Nadarajah, S. & Jameson, A. (2000). A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In *Aerospace Sciences Meetings*, American Institute of Aeronautics and Astronautics.

[101] Nadarajah, S. & Jameson, A. (2001). Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. In *15th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics.

[102] Nielsen, E.J. & Park, M.A. (2006). Using an Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design. *AIAA Journal*, **44**, 948–953.

[103] Nocedal, J. & Wright, S. (2006). *Numerical Optimization*. Springer.

[104] Özkaya, E. & Gauger, N. (2014). *One-shot methods for aerodynamic shape optimization*. Ph.D. thesis, RWTH Aachen University, Faculty of Mathematics, Computer Science and Natural Sciences.

[105] Özkaya, E. & Gauger, N.R. (2009). Single-step one-shot aerodynamic shape optimization. In K. Kunisch, J. Sprekels, G. Leugering & F. Tröltzsch, eds., *International Series of Numerical Mathematics*, vol. 158, 191–204, Birkhäuser Basel.

[106] Özkaya, E. & Gauger, N.R. (2010). Automatic transition from simulation to one-shot shape optimization with Navier-Stokes equations. *GAMM-Mitteilungen*, **33**, 133–147.

[107] Pacher, G., Pacher, H., Janeschitz, G., Kukushkin, A., Kotov, V. & Reiter, D. (2007). Modelling of DEMO core plasma consistent with

SOL/divertor simulations for long-pulse scenarios with impurity seeding. *Nuclear Fusion*, **47**, 469.

[108] Pacher, H., Kukushkin, A., Pacher, G., Kotov, V., Pitts, R. & Reiter, D. (2015). Impurity seeding in ITER DT plasmas in a carbon-free environment. *Journal of Nuclear Materials*, **463**, 591–595.

[109] Papoutsis-Kiachagias, E. & Giannakoglou, K. (2014). Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications. *Archives of Computational Methods in Engineering*, 1–45.

[110] Pironneau, O. (1974). On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, **64**, 97–110.

[111] Pitts, R., Buttery, R. & Pinches, S. (2006). Fusion: The way ahead. *Physics World*, **19**, 20–26.

[112] Powell, M. (1978). A fast algorithm for nonlinearly constrained optimization calculations. In G. Watson, ed., *Lecture Notes in Mathematics*, vol. 630, 144–157, Springer Berlin Heidelberg.

[113] Reiser, D. & Chandra, D. (2009). Plasma currents induced by resonant magnetic field perturbations in tokamaks. *Physics of Plasmas*, **16**.

[114] Reiter, D. (2009). *The Eirene Code User Manual, including: B2-EIRENE interface*.

[115] Reiter, D., Baelmans, M. & Börner, P. (2005). The EIRENE and B2-EIRENE codes. *Fusion Science and Technology*, **47**, 172–186.

[116] Rensink, M.E., Lodestro, L., Porter, G.D., Rognlien, T.D. & Coster, D.P. (1998). A Comparison of Neutral Gas Models for Divertor Plasmas. *Contrib. Plasma Phys.*, **38**, 325–330.

[117] Robinson, S.M. (1974). Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Mathematical programming*, **7**, 1–16.

[118] Rognlien, T., Milovich, J., Rensink, M. & Porter, G. (1992). A fully implicit, time dependent 2-D fluid code for modeling tokamak edge plasmas. *Journal of Nuclear Materials*, **196198**, 347–351.

[119] Romanelli, F., Laxåback, M. & JET EFDA Contributors (2011). Overview of JET results. *Nuclear Fusion*, **51**, 094008.

[120] Romanelli, F., Barabaschi, P., Borba, D., Federici, G., Horton, L. *et al.* (2012). A roadmap to the realization of Fusion Energy. Tech. rep., European Fusion Development Agreement (EFDA).

[121] Rozhansky, V., Voskoboynikov, S., Kaveeva, E., Coster, D. & Schneider, R. (2001). Simulation of tokamak edge plasma including self-consistent electric fields. *Nuclear Fusion*, **41**, 387.

[122] Ryutov, D. (2007). Geometrical properties of a snowflake divertor. *Physics of Plasmas (1994-present)*, **14**, 064502.

[123] Ryutov, D., Cohen, R., Rognlien, T. & Umansky, M. (2008). The magnetic field structure of a snowflake divertor. *Physics of Plasmas (1994-present)*, **15**, 092501.

[124] Ryutov, D., Cohen, R., Rognlien, T. & Umansky, M. (2012). A snowflake divertor: a possible solution to the power exhaust problem for tokamaks. *Plasma Physics and Controlled Fusion*, **54**, 124050.

[125] Schmitz, O., Evans, T.E., Fenstermacher, M.E., Frerichs, H., Jakubowski, M.W. *et al.* (2008). Aspects of three dimensional transport for ELM control experiments in ITER-similar shape plasmas at low collisionality in DIII-D. *Plasma Physics and Controlled Fusion*, **50**, 124029.

[126] Schneider, R., Bonnin, X., Borrass, K., Coster, D.P., Kastelewicz, H. *et al.* (2006). Plasma Edge Physics with B2-Eirene. *Contrib. Plasma Phys.*, **46**, 3–191.

[127] Schulz, V. & Gherman, I. (2009). One-Shot Methods for Aerodynamic Shape Optimization.

[128] Shafranov, V. (1958). On magnetohydrodynamical equilibrium configurations. *Soviet Journal of Experimental and Theoretical Physics*, **6**, 545.

[129] Shafranov, V.D. (1966). Plasma Equilibrium in a Magnetic Field. *Reviews of Plasma Physics*, **2**, 103.

[130] Simonini, R., Corrigan, G., Radford, G., Spence, J. & Taroni, A. (1994). Models and Numerics in the Multi-Fluid 2-D Edge Plasma Code EDGE2D/U. *Contrib. Plasma Phys.*, **34**, 368–373.

[131] Sokolowski, J. & Zochowski, A. (1999). On the Topological Derivative in Shape Optimization. *SIAM J. Control Optim.*, **37**, 1251–1272.

[132] Stambaugh, R., Lao, L. & Lazarus, E. (1992). Relation of vertical stability and aspect ratio in tokamaks. *Nuclear fusion*, **32**, 1642.

[133] Stangeby, P.C. (2000). *The plasma boundary of magnetic fusion devices*. Institute of Physics Publishing Bristol.

[134] Stangeby, P.C. (2000). The Role and Properties of the Sheath. In *Series in Plasma Physics*, 61–110, CRC Press.

[135] Stotler, D., Karney, C., Rensink, M. & Rognlien, T. (2000). Coupling of Parallelized DEGAS 2 and UEDGE Codes. *Contrib. Plasma Phys.*, **40**, 221–226.

[136] Ta'asan, S. (1995). Pseudo-time methods for constrained optimization problems governed by PDE. *ICASE report No. 95-32*.

[137] Taylor, T., Palacios, F., Duraisamy, K. & Alonso, J. (2012). Towards a Hybrid Adjoint Approach for Arbitrarily Complex Partial Differential Equations. In *Fluid Dynamics and Co-located Conferences*, American Institute of Aeronautics and Astronautics.

[138] Thévenin, D. & Janiga, G., eds. (2008). *Optimization and Computational Fluid Dynamics*. Springer Berlin Heidelberg.

[139] Tikhonov, A. & Arsenin, V. (1977). *Solutions of ill-posed problems*. Winston, Washington.

[140] Tikhonov, A.N. (1963). Solution of incorrectly formulated problems and the regularization method. *Doklady Akademii Nauk SSSR*, **151**, 501–504.

[141] Tröltzsch, F. (2010). *Optimal Control of Partial Differential Equations: Theory, Methods, and Applications*. American Mathematical Soc., Providence, Rhode Island.

[142] Turnyanskiy, M., Neu, R., Albanese, R., Ambrosino, R., Bachmann, C. *et al.* (2015). European roadmap to the realization of fusion energy: Mission for solution on heat-exhaust systems. *Fusion Engineering and Design*, **96-97**, 361–364, proceedings of the 28th Symposium On Fusion Technology (SOFT-28).

[143] Valanju, P., Kotschenreuther, M. & Mahajan, S. (2010). Super X divertors for solving heat and neutron flux problems of fusion devices. *Fusion Engineering and Design*, **85**, 46–52.

[144] Van den Kerkhof, S. (2016). *Plasma Edge Simulation of Innovative Magnetic Topologies in Tokamak Devices*. Master's thesis, Department of mechanical engineering, KU Leuven.

[145] Van Oevelen, T. & Baelmans, M. (2014). *Optimal Heat Sink Design for Liquid Cooling of Electronics (Optimaal ontwerp van koellichamen voor vloeistofkoeling van elektronica)*. Ph.D. thesis, KU Leuven.

[146] Viola, B., Frigione, D., Belo, P., Groth, M., Kempenaars, M. *et al.* (2013). Study of the Effect of the Outer-Strike Point Location on the Divertor Neutral Pressure in JET-ILW Using EDGE2D/EIRENE. In *40th European Physical Society Conference on Plasma Physics*, European Physical Society, Espoo, Finland.

[147] Walther, A., N. R. Gauger, L.K. & Richert, N. (2016). On an extension of one-shot methods to incorporate additional constraints. *Optimization Methods and Software*.

[148] Wesson, J. (1978). Hydromagnetic stability of tokamaks. *Nuclear Fusion*, **18**, 87.

[149] Wesson, J. & Campbell, D.J. (2011). *Tokamaks*, vol. 149. Oxford University Press.

[150] Wiesen, S. (2006). EDGE2D/EIRENE code interface report. Tech. rep., Forschungszentrum Juelich.

[151] Wiesen, S., Reiter, D., Kotov, V., Baelmans, M., Dekeyser, W. *et al.* (2015). The new SOLPS-ITER code package. *Journal of Nuclear Materials*, **463**, 480–484.

Curriculum vitae

Maarten Blommaert Born June 27, 1989 in Brasschaat, Belgium

Current position

From 04/2016 Researcher
Department of Mechanical Engineering
KU Leuven, Belgium

Former position

2012 - 2016 Researcher PhD student
Plasmaphysics Division (IEK-4)
Forschungszentrum Jülich, Germany

University education

2007 - 2010 Bachelor of Science in Engineering
Mechanical and Electrical Engineering
KU Leuven, Belgium

2010 - 2012 Master of Science in Engineering
Thermomechanical Energy Engineering
KU Leuven, Belgium

Awards

2013 2nd poster price
Aachen Conference on Computational Engineering Science

Experience

2013 - 2016 Tutor of 6 MSc in Engineering theses (KU Leuven)
on numerical simulation and optimization
in plasma edge modelling

2012 - 2016 Aerodynamics: lab sessions (Master course, KU Leuven)

07-08/2014 CEMRACS Research stay at CIRM, Luminy, France
Project: CEDRES++ for Automated Divertor Design

Memberships

HITEC Graduate School for Energy- and Climate Research

OPTEC KU Leuven Center of Excellence:
Optimization in Engineering

AICES Aachen Institute for Advanced Study
in Computational Engineering Science

Publications

Articles in internationally reviewed academic journals

1. Baelmans, M., Blommaert, M., De Schutter, J., Dekeyser, W. & Reiter, D. (2014). Efficient parameter estimation in 2D transport models based on an adjoint formalism. *Plasma Physics and Controlled Fusion*, **56**, 114009.

2. Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N. & Reiter, D. (2014). An Automated Approach to Magnetic Divertor Configuration Design. *Nuclear Fusion*, **55**.

3. Blommaert, M., Baelmans, M., Dekeyser, W., Gauger, N. & Reiter, D. (2015). A novel approach to magnetic divertor configuration design. *Journal of Nuclear Materials*, **463**, 1220–1224.

4. Blommaert, M., Heumann, H., Baelmans, M., Gauger, N.R. & Reiter, D. (2016). Towards Automated Magnetic Divertor Design for Optimal Heat Exhaust. *ESAIM: Proceedings and Surveys*, **53**, 49–63.

5. Blommaert, M., Baelmans, M., Heumann, H., Marandet, Y., Bufferand, H. *et al.* (2016). Magnetic Field Models and their Application in Optimal Magnetic Divertor Design. *Contributions to Plasma Physics*, **56**, 796–801.

6. Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N.R. & Reiter, D. (2017). A practical globalization of one-shot optimization for optimal design of tokamak divertors. *Journal of Computational Physics*, **328**, 399–412.

7. Baelmans, M., Blommaert, M., Dekeyser, W., Van Oevelen, T., & Reiter, D. (2016). Achievements and Challenges in Automated Parameter, Shape and Topology Optimization for Divertor Design. *Nuclear Fusion*, **57**, 036022.

8. Blommaert, M., Reiter, D. & Baelmans, M. (2016). An Efficient Methodology to Analyze Plasma Edge Model Parameter Sensitivities. *Nuclear Materials and Energy*, in press.

Meeting abstracts, presented at conferences and symposia

1. Blommaert, M., Baelmans, M., & Reiter, D. (2016). An Efficient Methodology to Analyze Plasma Edge Model Parameter Sensitivities (poster). *22nd International Conference on Plasma Surface Interactions in Controlled Fusion Devices*, Rome, Italy, 30 May - 3 June 2016.

2. Blommaert, M., Baelmans, M., Marandet, Y., Bufferand, H. *et al.* (2016). Magnetic Field Models and their Application in Optimal Magnetic Divertor Design (poster). *15th International Workshop on Plasma Edge Theory in Fusion Devices*, Nara, Japan, 9 - 11 September 2015.

3. Baelmans, M., Blommaert, M., Dekeyser, W., Van Oevelen, T., & Reiter, D. (2015). Achievements and Challenges in Automated Parameter, Shape and Topology Optimization for Divertor Design (presentation). *1st IAEA Technical Meeting on Divertor Concepts*, Aachen, 29 September - 02 October 2015.

4. Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N.R., & Reiter, D. (2015). A practical and in parts adjoint based gradient computation methodology for efficient optimal magnetic divertor design in nuclear fusion reactors (presentation). *2nd Frontiers of Computational Physics*, Zurich, Switzerland, 3-5 June 2015.

5. Blommaert, M., Heumann, H., Baelmans, M., Gauger, N., & Reiter, & D. (2015). An automated approach to magnetic divertor configuration design, using an efficient optimization methodology (presentation). *79th Annual Meeting of the DPG and DPG Spring Meeting*, Bochum, 2-5 March 2015.

6. Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N., & Reiter, D. (2014). Towards an automated approach to magnetic divertor configuration design (presentation). *78th Annual Meeting of the DPG and DPG Spring Meeting*, Berlin, 17-21 March 2014.

7. Baelmans, M., Blommaert, M., Dekeyser, W., Gauger, N., & Reiter, D. (2014). A novel approach to magnetic divertor configuration design (poster). *21st International Conference on Plasma Surface Interactions*, Kanazawa, Japan, 26-30 May 2014.

8. Baelmans, M., De Schutter, J., Dekeyser, W., Blommaert, M., & Reiter, D. (2013). Efficient parameter estimation in 2D transport models based on an adjoint formalism (presentation). *8th workshop on Fusion Data Processing, Validation and Analysis*, Ghent, Belgium, 04-06 November 2013.

9. Blommaert, M., Dekeyser, W., Baelmans, M., Gauger, N., & Reiter, D. (2013). An automated approach to magnetic divertor configuration design (poster). *AC.CES 2013*, Aachen, Germany, 09-11 September 2013.

Band / Volume 353
Untersuchungen zur Luftqualität in Bad Homburg
C. Ehlers, D. Klemp, C. Kofahl, H. Fröhlich, M. Möllmann-Coers und A. Wahner (2016), IV, 93 pp
ISBN: 978-3-95806-199-6

Band / Volume 354
Herstellung thermisch gespritzter Schichten mit optimierten Spannungseigenschaften
M. Mutter (2016), VI, 142, VII-XXII, xxvi pp
ISBN: 978-3-95806-200-9

Band / Volume 355
Entwicklung selbstheilender Wärmedämmschichten
D. Koch (2016), X, 120 pp
ISBN: 978-3-95806-201-6

Band / Volume 356
Betriebsstrategien für Brenngaserzeugungssysteme zur Anwendung in HT-PEFC-Hilfsstromaggregaten
D. Krekel (2017), IX, 265 pp
ISBN: 978-3-95806-203-0

Band / Volume 357
Korrosion metallischer Bipolarplatten in Hochtemperatur-Polymerelektrolyt-Brennstoffzellen
V. Weißbecker (2017), viii, 194 pp
ISBN: 978-3-95806-205-4

Band / Volume 358
Realistic Bandwidth Estimation in the Theoretically Predicted Radionuclide Inventory of PWR-UO2 Spent Fuel Derived from Reactor Design and Operating Data
I. Fast (2017), XI, 129 pp
ISBN: 978-3-95806-206-1

Band / Volume 359
Light Trapping by Light Treatment
Direct Laser Interference Patterning
For the Texturing of Front Contacts in Thin-Film Silicon Solar Cells
T. Dyck (2017), vi, 172, XI pp
ISBN: 978-3-95806-208-5

Band / Volume 360
Interface and Topography Optimization for Thin-Film Silicon Solar Cells with Doped Microcrystalline Silicon Oxide Layers
C. Zhang (2017), VII, 156 pp
ISBN: 978-3-95806-209-2

Band / Volume 361
Calibration and large-scale inversion of multi-configuration electromagnetic induction data for vadose zone characterization
C. von Hebel (2017), ix, 123 pp
ISBN: 978-3-95806-210-8

Band / Volume 362
Process-based modelling of regional water and energy fluxes taking into account measured neutron intensities by cosmic-ray probes
R. Baatz (2017), xvi, 135 pp
ISBN: 978-3-95806-211-5

Band / Volume 363
Quantitative Analyse der Lithiumverteilung in Kathoden- und Elektrolyt-Dünnschichten für Festkörperbatterien
C. Dellen (2017), vi, 161 pp
ISBN: 978-3-95806-214-6

Band / Volume 364
Intragruppentrennung Seltener Erden mittels neuer phosphororganischer Liganden
S. Hadić (2017), VIII, 164 pp
ISBN: 978-3-95806-215-3

Band / Volume 365
Automated Magnetic Divertor Design for Optimal Power Exhaust
M. Blommaert (2017), xxiv, 219 pp
ISBN: 978-3-95806-216-0

Energie & Umwelt /
Energy & Environment
Band / Volume 365
ISBN 978-3-95806-216-0