RICE UNIVERSITY

Trust-Region SQP Methods with Inexact Linear System Solves for Large-Scale Optimization

by

Denis Ridzal

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Matthias Heinkenschloss, Chairman
Professor of Computational and Applied Mathematics

William W. Symes
Noah Harding Professor of Computational and Applied Mathematics

Timothy C. Warburton
Assistant Professor of Computational and Applied Mathematics

Matteo Pasquali
Associate Professor of Chemical and Biomolecular Engineering

Houston, Texas
April, 2006
Abstract

Trust-Region SQP Methods with Inexact Linear System Solves for Large-Scale Optimization

by

Denis Ridzal

This thesis extends the design and the global convergence analysis of a class of trust–region sequential quadratic programming (SQP) algorithms for smooth nonlinear optimization to allow for an efficient integration of inexact linear system solvers.

Each iteration within an SQP method requires the solution of several linear systems, whose system matrix/operator involves the linearized constraints. Most existing implementations of SQP algorithms use direct linear algebra methods to solve these systems. For many optimization problems in science and engineering this is infeasible, because the systems are too large or the matrices associated with the linearized constraints are not formed explicitly. Instead, iterative solvers, such as preconditioned Krylov subspace methods have to be applied for the approximate solution of the linear systems arising within the SQP algorithm. In this case, the optimization algorithm has to provide stopping tolerances for the iterative solver.

The existing literature on the treatment of inexact linear system solves in SQP algorithms is rather scarce. Most theoretical results either provide stopping tolerances for iterative solvers that cannot be easily implemented in practice, or are restricted to specific classes of optimization problems. This thesis provides concrete stopping criteria for the iterative solution of so-called augmented systems, which allows for
a wider applicability of the resulting SQP algorithm and a rigorous integration of available KKT preconditioners. A key contribution is the development of an inexact conjugate gradient algorithm for the solution of quadratic subproblems with linear constraints that are subject to arbitrary nonlinear perturbations that arise from the approximate computation of projections via Krylov subspace methods.

The resulting SQP algorithm dynamically adjusts stopping tolerances for iterative linear system solves based on its current progress toward a KKT point. The stopping tolerances can be easily implemented and efficiently computed, and are sufficient to guarantee first-order global convergence of the algorithm. The performance of the algorithm is examined on optimal control problems governed by Burgers and Navier–Stokes equations.
Acknowledgements

First, I would like to thank my advisor, Matthias Heinkenschloss, for his support and patience, guidance when I needed it, and infusions of enthusiasm during the rough stretches. I also wish to thank the rest of my committee for their interest in the topic and their appreciation for the difficulty of the issues that I attempted to tackle.

The beginning of this five–year journey was the doing of my undergraduate advisor, Ray Chin, who introduced me to the world of computational and applied mathematics, and recommended Rice University for my graduate studies. For this, I am forever indebted to him.

I would also like to thank Eddie, Fernando, Michael, John, and Dwayne for watching my back, and letting me return the favor. Many thanks to the fellow thirty–sixteeners, Hoang, Agata, Andreas, Patricia, and Kary, for providing an inspirational and productive working environment.

For friendship, love, and ultimate patience, I thank Heidi Thornquist.

Most importantly, I wish to thank my parents, Nada and Muharem Ridzal, for always believing in me. They taught me to be tough, work hard, and never give up, skills without which this work could have never been completed. I thank them for their constant support and their love.

This research was supported by NSF grant ACI–0121360.
Contents

Abstract ii

Acknowledgements iv

List of Figures viii

List of Tables ix

1 Introduction 1

1.1 Existing Work 2

1.1.1 Available SQP Implementations 2

1.1.2 The Issue of Inexactness in Optimization Algorithms 4

1.1.3 Shortcomings of the Current Approaches to Inexactness Control 6

1.2 Organization of the Thesis 7

2 The Composite–Step Trust–Region SQP Method 8

2.1 Analysis of the Trust-Region SQP Algorithm 8

2.1.1 Examples of Null–Space Representations \( W_k \) 11

2.1.2 Lagrange Multipliers 13

2.1.3 Hessian Approximations 14

2.1.4 Solution of the Quadratic Subproblem 14

2.1.5 Acceptance of the Trial Step and the Update of the Trust–Region Radius 24
# 3 Inexactness in a Generalized Trust–Region SQP Algorithm

3.1 Inexactness in the Quasi–Normal Step

3.2 Inexactness in the Tangential Step

3.3 Balancing Progress in the Computation of the Tangential and the Quasi–Normal Step

3.4 Statement of the Generalized Trust–Region SQP Algorithm with Inexactness Control

3.5 Global Convergence of Alg. 3.4.2

3.6 A Generalized Conjugate Gradient Method for the Solution of the Tangential Subproblem

# 4 Inexact Linear System Solves in an SQP Algorithm with Null–Space Projections

4.1 Tangential Step

4.1.1 Satisfaction of General Convergence Requirements for Inexactly Computed Tangential Steps

4.1.2 Practical Convergence Requirements for Inexactly Computed Tangential Steps

4.2 Quasi–Normal Step

4.3 Lagrange Multipliers

4.4 Balancing the Tangential and the Quasi–Normal Step

4.5 Statement of the Trust–Region SQP Algorithm with Inexact Augmented System Solves

# 5 Numerical Examples

5.1 Optimal Control of Burgers Equation in One Dimension

5.1.1 Problem Formulation

5.1.2 Problem Discretization

5.1.3 Numerical Results
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2 Optimal Control of Navier–Stokes Equations in Two Dimensions</td>
<td>99</td>
</tr>
<tr>
<td>5.2.1 Problem Formulation</td>
<td>99</td>
</tr>
<tr>
<td>5.2.2 Numerical Results</td>
<td>103</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>111</td>
</tr>
<tr>
<td>Bibliography</td>
<td>113</td>
</tr>
</tbody>
</table>
List of Figures

2.1 The composite–step approach ................................................. 15

3.1 Interaction between the quasi–normal and the tangential step. .... 29

4.1 Balancing the quasi–normal and the effective tangential step. ..... 79

5.1 The matrix $N'(\vec{u})$. ................................................. 91
5.2 Absolute GMRES stopping tolerances – Burgers. ................. 94
5.3 Relative GMRES stopping tolerances – Burgers. ................... 95
5.4 Performance of preconditioned GMRES – Burgers. ............... 96
5.5 Geometry of the backward–facing step channel. .................... 99
5.6 The finite element grid for the backstep channel. ................. 104
5.7 The velocity plot for the uncontrolled fluid flow. ................. 105
5.8 Uncontrolled flow: recirculation near the corner region. ........ 105
5.9 The velocity plot for the controlled fluid flow. .................... 106
5.10 Absolute GMRES stopping tolerances – Navier–Stokes. ........ 106
5.11 Relative GMRES stopping tolerances – Navier–Stokes. ........ 107
5.12 Performance of preconditioned GMRES – Navier–Stokes. ...... 108
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Inexactness control vs. fixed stopping tolerances – Burgers.</td>
<td>95</td>
</tr>
<tr>
<td>5.2</td>
<td>SQP convergence history with inexactness control – Burgers.</td>
<td>97</td>
</tr>
<tr>
<td>5.3</td>
<td>SQP convergence history with exact solves – Burgers.</td>
<td>98</td>
</tr>
<tr>
<td>5.4</td>
<td>Inexactness control vs. fixed stopping tolerances – Navier–Stokes.</td>
<td>107</td>
</tr>
<tr>
<td>5.5</td>
<td>SQP convergence history with inexactness control – Navier–Stokes.</td>
<td>109</td>
</tr>
<tr>
<td>5.6</td>
<td>SQP convergence history with exact solves – Navier–Stokes.</td>
<td>110</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The sequential quadratic programming (SQP) methodology represents the state of the art for the solution of smooth nonconvex, nonlinear optimization problems. In combination with effective strategies for the treatment of inequality constraints, it is used in powerful optimization codes for the solution of a wide range of nonlinear optimization problems. For example, the popular optimization codes SNOPT, KNITRO and LOQO utilize the SQP methodology.

Each iteration within an SQP algorithm requires the solution of several linear systems, whose system matrix/operator involves the linearized constraints. Most existing implementations of SQP algorithms, including those mentioned above, use direct linear algebra methods to solve these systems. Lately though, serious efforts have been made in applying the SQP methodology to large–scale optimization problems within parallel computing environments. In this scenario, iterative solvers, such as preconditioned Krylov methods have to be applied for the approximate solution of linear systems arising in the SQP algorithm.

Iterative linear system solvers are also required for problems whose structure does not allow for an explicit representation of linear operators, as is the case for many optimization problems in science and engineering. In addition, the use of specialized linear system solvers, such as multigrid or domain decomposition methods, is often required in optimization problems governed by partial differential equations (PDEs).
The use of iterative solvers results in several complications. First, global convergence theories for most concrete SQP schemes assume that linear systems can be solved exactly. Iterative solvers, however, are inherently inexact. Our task will be to examine how the global convergence properties of SQP algorithms are affected by inexactness in linear system solves and to determine generic global convergence conditions.

The second, more difficult task is to let the optimization algorithm dynamically determine stopping tolerances for iterative solvers. Here we insist on only using stopping tolerances that are easily implementable and that can be efficiently computed in practice. The computed tolerances must be fairly loose when the SQP iterate is far away from the solution, but sufficient to guarantee global convergence of the SQP algorithm. As SQP iterates approach the solution, the stopping tolerances for linear system solves must be lowered accordingly.

1.1 Existing Work

The purpose of this section to familiarize the reader with a representative subset of SQP algorithms and corresponding codes, the available work on inexact linear system solves in SQP algorithms (and the general issue of inexactness in optimization), and the shortcomings of the current approaches to inexactness control.

1.1.1 Available SQP Implementations

Strong convergence properties of SQP algorithms combined with their robustness make them successful methods for the solution of nonlinearly constrained optimization problems. Of course, efficient solution of the underlying linear systems plays an important role as well. Early SQP codes were based on dense linear algebra, while the codes of the 1990s rely on sparse direct solution techniques. On the other hand, some more recent implementations can utilize iterative linear solvers.
One of the earliest codes, based on dense linear algebra, is NPSOL, developed in the 1980s by Gill, et al. [40, 41]. To ensure convergence from remote starting points, line search is used, while inequality constraints are treated by an active–set approach. The applicability of NPSOL to large–scale applications is intrinsically limited by the use of dense linear algebra. Another SQP code utilizing dense linear algebra is NLPQL, developed in the early 1980’s by Schittkowski [67].

In the mid-1990s Gill, et al. [42] developed SNOPT, one of the first commercially available SQP codes to utilize sparse direct linear algebra. Like NPSOL, SNOPT is a line search method with an active–set approach for the treatment of inequality constraints. An entirely different approach is implemented in KNITRO, developed by Byrd, et al. [19, 75, 20], where inequality constraints are handled via an interior–point method. Linear systems are solved using sparse LU factorizations. Yet another interior–point code is LOQO, developed by Vanderbei, et al. [68, 73]. At the lowest level, LOQO solves modified KKT systems using sparse Cholesky factorizations. Other SQP codes based on sparse direct linear algebra include SPRNLP, developed by Betts and Frank [9, 10] and used in the SOCS optimal control framework, and IPOPT, developed by Wächter, et al. [74].

This concludes our review of codes based on direct (dense or sparse) linear algebra. For an excellent overview of the history and practice of SQP methods from their first emergence in 1963 (Wilson [76]) to the developments of the early 1990s, we refer the reader to Boggs and Tolle [13]. An exhaustive theoretical study of modern algorithms is given by Gould and Toint [48] as well as Conn, Gould, and Toint [22, Ch.15], whereas a comparative performance evaluation of the latest implementations is provided by Benson, et al. [7]. Additional details about the above codes can be found on the NEOS server for optimization [23].

The reviewed methods have been effectively used in applications ranging from engineering to finance, and have greatly contributed to the popularization and success of SQP methods. On the other hand, their limitations are evident. They always tie
the underlying SQP algorithm to a specific linear algebra engine. This makes the use of problem–specific solvers impossible. All exclusively use direct linear algebra, which due to memory limitations, cannot be applied to large-scale problems involving, for example, PDE simulations of complex physical systems.

We now turn our attention to the existing work in the area of matrix–free SQP codes. MOOCHO, developed by Bartlett [6], is an object–oriented framework for the solution of large–scale nonlinear optimization problems using gradient–based methods. MOOCHO evolved from rSQP++, an implementation of a reduced–space SQP algorithm [3]. One of the most attractive features of the MOOCHO framework is that it supports fully abstract linear algebra, allowing complex problem simulations on parallel distributed–memory supercomputers. A disadvantage of the underlying optimization algorithm is that it does not take into account the issue of inexactness arising from iterative linear system solves. Linear systems are solved to a fixed stopping tolerance, which usually results in their oversolving.

One currently available SQP code that is built around a theoretical framework for the treatment of inexact linear system solves is TRICE, developed by Heinkenschloss, et al. [29, 55, 54]. TRICE is an interior–point, trust–region SQP code that uses a reduced Hessian approach for the formulation of quadratic subproblems. It is primarily geared toward the solution of optimal control problems, in which there is a clear separation between a state (basis) variable space and a control (design, non–basis) space. TRICE lacks a full integration of practical mechanisms for inexactness control in linear system solves. Consequently, no numerical results documenting their performance are available.

1.1.2 The Issue of Inexactness in Optimization Algorithms

There is a considerable amount of literature available on the subject of inexactness in the context of Newton methods for unconstrained optimization and nonlinear equations. A series of publications related to the treatment of local convergence of inexact
Newton methods for unconstrained optimization dates back to the early 1980s, see, for example, Dembo, et al. \cite{24,25} and Dennis and Walker \cite{31}. The connection with inexact SQP methods is established shortly thereafter, by Dembo and Tulowitzki \cite{26,27} and Fontecilla \cite{38}, although only on the level of local convergence analysis.

An early local convergence result for inexact Newton methods for nonlinear equations, due to Ypma, can be found in \cite{77}.

Global convergence behavior of inexact trust–region algorithms for unconstrained optimization is investigated by Carter in \cite{21}, while the first global convergence results in the context of nonlinear equations were obtained by Deuflhard \cite{32}, Eisenstat and Walker \cite{33,34}, and Brown and Saad \cite{17}.

The first global convergence result for an SQP method is in the work of Jäger and Sachs \cite{59}, where they investigate a line–search reduced–space approach. They, however, do not give an implementable mechanism for the control of inexactness, due to the need to compute Lipschitz constants in order to establish certain bounds. In a similar fashion, Biros and Ghattas \cite{11} establish convergence requirements for an inexact quasi-Newton reduced-space SQP method. Their approach requires tight bounds on derivative norms, which are hard to estimate in practice.

One known result that gives practical, easily enforceable accuracy requirements is due to Heinkenschloss and Vicente \cite{55}. They focus on a composite–step trust-region SQP technique in the reduced space, and manage to effectively control inexactness in first–order derivative information as well as inexact linearized constraint equation solves. Their algorithm, albeit tied to the reduced-space approach, represents an excellent starting point for our discourse on inexactness in the full-space setting, in Chapter \ref{chap:3}.

A recent study by Ulbrich \cite{72} treats a generalized composite–step SQP method in the context of time–dependent PDE–constrained optimization. The SQP algorithm is based on a nonmonotone trust–region method without a penalty function, and offers a unique way of balancing feasibility and optimality. Some of the inexactness handling
mechanisms are similar to the inexactness framework of Heinkenschloss and Vicente [55]. Ulbrich’s framework is suited for PDE–constrained optimization problems that rely on separate state and adjoint equation solves.

1.1.3 Shortcomings of the Current Approaches to Inexactness Control

As presented above, there are essentially only two theoretical studies of inexactness in SQP algorithms that are satisfactory from the point of view of implementability and computational efficiency of the devised stopping criteria for iterative linear system solves, [55, 72]. All others fall short of this goal, due to the need for estimates of Lipschitz constants and norms of derivative operators.

The approaches of Heinkenschloss and Vicente, and Ulbrich, however, have certain shortcomings. Both are tied to reduced–space SQP methods that rely closely on what is known as the basis–nonbasis decomposition of optimization variables. We consider a more general, full–space composite–step approach, in which this distinction is not necessary.

The second shortcoming of [55, 72] is in the lack of treatment of concrete algorithms for the computation of the trial step in composite–step trust–region methods. A trial step in these algorithms is composed as the sum of two sub–steps, whose computation requires the solution of several linear systems. One of these steps, known as the tangential step, is usually computed using a conjugate gradient method, and requires many linear system solves, often dominating the computational cost of the entire SQP algorithm. In [55, 72], the computation of the tangential step is not considered in sufficient detail. Our key contribution in the tangential step computation is the development of an inexact conjugate gradient algorithm that utilizes a series of approximate projections via Krylov subspace methods, whose accuracy is controlled by the SQP algorithm. The computation of approximate projections is related to the iterative solution of so called augmented systems, which allow for a wider appli-
cability of the resulting SQP algorithm and a rigorous integration of available KKT preconditioners.

1.2 Organization of the Thesis

Chapter 2 gives a brief overview of the composite-step trust-region SQP methodology and states the relevant global convergence conditions and results. The convergence requirements stated in this chapter will guide our analysis of inexactness, which is presented in Chapter 3. In this chapter, we identify the key components of the SQP algorithm that are affected by inexact linear system solves, and slightly generalize the framework for SQP methods with inexact linear system presented in [55] to handle general null-space representations. In Chapter 4 we present a concrete implementation of the SQP method using null-space projections and show how this implementation fits into the inexactness framework of Chapter 3. Chapter 5 contains numerical results, which demonstrate the efficiency of the devised mechanisms for inexactness control in linear system solves. Finally, Chapter 6 summarizes the highlights of our work and discusses potential extensions.
Chapter 2

The Composite–Step Trust–Region SQP Method

In this chapter we describe the basic structure of an SQP algorithm for equality-constrained optimization. We are particularly interested in algorithms that employ trust-region methods, due to the proven robustness of this globalization strategy. It should be noted that the theoretical foundations reviewed in this chapter serve merely as a stepping stone toward the inexactness-controlling schemes proposed in Chapter 3. Thus the primary purpose of this chapter is to familiarize the reader with our SQP approach, present elementary theoretical results, and introduce the core algorithmic modules.

The equality-constrained problem is discussed in Section 2.1. It is solved using a composite-step trust-region approach in the full variable space. Along with concrete algorithms, we will present the basic global convergence requirements and results.

2.1 Analysis of the Trust-Region SQP Algorithm

We consider the following generalized equality-constrained nonlinear programming problem (NLP):

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad c(x) = 0,
\end{align*}
\]

(2.1a)

(2.1b)
where \( f : \mathcal{X} \to \mathbb{R} \) and \( c : \mathcal{X} \to \mathcal{Y} \) for some Hilbert spaces \( \mathcal{X} \) and \( \mathcal{Y} \). To derive an SQP model for the problem (2.1), we proceed as follows.

First, we present the basic building blocks of the standard KKT theory, and introduce some notation. We assume that \( f \) and \( c \) are twice continuously differentiable with Lipschitz continuous second derivatives. It is necessary to define the Lagrangian functional \( \mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \) for (2.1), given by the expression

\[
\mathcal{L}(x, \lambda) = f(x) + \langle \lambda, c(x) \rangle.
\]

If \( x_* \) satisfies the constraint equation (2.1b) and if \( c_x(x_*) \) is surjective, then \( x_* \) is called a regular point. If additionally \( x_* \) is a local solution of (2.1), then there exists a Lagrange multiplier \( \lambda_* \in \mathcal{Y} \) such that the first order necessary optimality conditions

\[
\begin{align*}
\nabla_x f(x_*) + c_x(x_*)^* \lambda_* &= 0 \quad (2.2a) \\
c(x_*) &= 0 \quad (2.2b)
\end{align*}
\]

are satisfied. The system (2.2) is also known as the KKT system. Finally, if \( x_*, \lambda_* \) satisfy (2.2), and

\[
\nabla_{xx} \mathcal{L}(x_*, \lambda_*)[h, h] \geq \rho \|h\|^2 \quad \forall h \in \text{Null}(c_x(x_*)),
\]

(2.3)

for some \( \rho > 0 \), then \( x_* \) is a strict local minimum of (2.1). This statement represents the second order sufficient optimality conditions.

If Newton’s method is used for the solution of the KKT system (2.2), then at the \( k \)-th iterate \( (x_k, \lambda_k) \) the following system must be solved:

\[
\begin{pmatrix}
\nabla_{xx} \mathcal{L}(x_k, \lambda_k) & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
s_k^x \\
s_k^\lambda
\end{pmatrix} = - \begin{pmatrix}
\nabla_x f(x_k) + c_x(x_k)^* \lambda \\
c(x_k)
\end{pmatrix}.
\] (2.4)

The foundation of the SQP methodology is the fact that the Newton system (2.4)
can be interpreted as the KKT system for the solution of the quadratic problem

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle \nabla_{xx} \mathcal{L}(x_k, \lambda_k) s^x_k, s^x_k \rangle_X + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s^x_k \rangle_X + \mathcal{L}(x_k, \lambda_k) \\
\text{s.t.} & \quad c_x(x_k) s^x_k + c(x_k) = 0,
\end{align*}
\]

assuming that there exists a \( \rho_k > 0 \) such that \( \nabla_{xx} \mathcal{L}(x_k, \lambda_k)[h, h] \geq \rho \|h\|^2 \), for all \( h \in \text{Null}(c_x(x_k)) \). In other words, the SQP approach amounts to repeatedly minimizing a quadratic approximation of the Lagrangian with respect to a linearization of the constraints. In order to achieve global convergence, we further constrain the step by a trust-region bound. At each iterate \( x_k \), this yields the problem

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle H_k s^x_k, s^x_k \rangle_X + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s^x_k \rangle_X + \mathcal{L}(x_k, \lambda_k) \\
\text{s.t.} & \quad c_x(x_k) s^x_k + c(x_k) = 0 \\
& \quad \|s^x_k\|_X \leq \Delta_k,
\end{align*}
\]

where \( \Delta_k \) is a given trust-region radius. At this point, it is important to note two things. First, we have introduced a new operator \( H_k = H(x_k, \lambda_k) \) that in some sense “approximates” the exact Hessian of the Lagrangian \( \nabla_{xx} \mathcal{L}(x_k, \lambda_k) \). Section 2.1.3 discusses this modification briefly. Second, one should be aware of the fact that the constraints (2.6b)-(2.6c) in general might not be consistent, and that an algorithm for the solution of (2.6) must take this into account. Once the step \( s^x_k \) has been computed, if it gives sufficient decrease for a chosen reduction measure (this remains to be discussed in Section 2.1.5), it is accepted, otherwise, it is rejected, and the trust region radius is reduced.

While this covers the basics of trust-region SQP, we have not discussed the underlying computational algorithms yet. These can, of course, vary from one implementation to another. In general, we have to focus on four main components: computation of Lagrange multipliers, computation of Hessian approximations, solu-
tion of the trust-region QP problem (2.6), and acceptance of the trial step $s_k^x$. The theoretical requirements on each of these modules and the corresponding statement of convergence for our SQP scheme are based on the following general assumptions, as stated in [28] and modified to fit the Hilbert space setting:

(A1) For all iterations $k$, we assume that $x_k \in \Omega$ and $x_k + s_k^x \in \Omega$, where $\Omega$ is an open subset of $\mathcal{X}$.

(A2) The function(al)s $f$ and $c$ are twice continuously Fréchet differentiable with Lipschitz continuous second derivatives.

(A3) The operator $c_x(x)$ is surjective for all $x \in \Omega$.

(A4) The following function(al)s and operators are uniformly bounded over all $x \in \Omega$: $f(x)$, $\nabla_x f(x)$, $\nabla_{xx} f(x)$, $c(x)$, $c_x(x)$, $(c_x(x)c_x(x)^*)^{-1}$, and $c_{xx}(x)$.

(A5) For every iterate $x_k$, there exists a bounded linear operator $W_k : \mathcal{Z} \rightarrow \mathcal{X}$, where $\mathcal{Z}$ is a Hilbert space, such that

$$\text{Range}(W_k) = \text{Null}(c_x(x_k)).$$

Moreover, the sequence of operators $W_k$, $k \in \mathbb{N}$, is bounded.

(A6) The sequence of operators $H_k$, $k \in \mathbb{N}$, is bounded.

(A7) The sequence of Lagrange multipliers $\lambda_k$, $k \in \mathbb{N}$, is bounded.

## 2.1.1 Examples of Null–Space Representations $W_k$

It is very important to note that the null–space operator $W_k$ introduced in the general assumption (A5) is not a mere theoretical construct. It represents a generic tool for handling the linear constraint $c_x(x_k)s_k^x + c(x_k) = 0$, given in (2.6b). Several concrete approaches can be found in the literature on SQP methods. We discuss them next.
The QR Approach

If $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \mathbb{R}^m$, $m \leq n$, then one can use the QR decomposition of the linearized constraints to compute $W_k$, see, e.g., [43]. Assume that $c_x(x_k) \in \mathbb{R}^{m \times n}$, $c_x(x_k)$ has full row rank $m$. We have

$$c_x(x_k)^* = Q_k R_k = \begin{bmatrix} Q_k^1 & Q_k^2 \\ 0 & \end{bmatrix} \begin{bmatrix} R_k^1 \\ 0 \end{bmatrix},$$

(2.7)

where $Q_k \in \mathbb{R}^{n \times n}$ is orthogonal, $Q_k^1 \in \mathbb{R}^{n \times m}$ represents the first $m$ columns of $Q_k$, $Q_k^2 \in \mathbb{R}^{n \times (n-m)}$ represents the remaining $n - m$ columns of $Q_k$, and $R_k^1 \in \mathbb{R}^{m \times m}$ is upper triangular. In this case the the domain of the null–space representer $W_k$ is $Z = \mathbb{R}^{n-m}$. A simple argument shows that the columns of the matrix

$$W_k = Q_k^2$$

form a basis for the null space of $c_x(x_k)$.

The Basis–Nonbasis Approach

A different null–space representation is often used in, e.g. PDE–constrained optimization problems, in which there is a clear separation between basis (state) and nonbasis (control, design) variables. The optimization problem can be formulated as

$$\begin{align*}
\min & \quad f(u, g) \\
\text{s.t.} & \quad c(u, g) = 0,
\end{align*}$$

(2.8a)

(2.8b)

where $f : U \times G \to \mathbb{R}$ and $c : U \times G \to U$ for some Hilbert spaces $U$ and $G$. In (2.8) $u$ represent the states and $g$ represent the control/design variables. We assume that $c_u(u, g)$ is invertible. In this case the variable space is $\mathcal{X} = U \times G$, the constraint space is $\mathcal{Y} = U$, and the domain of $W_k$ is $Z = G$. The operator representing the null
space of the linearized constraints is given by \( W_k : \mathcal{G} \rightarrow \mathcal{U} \times \mathcal{G} \), defined by

\[
W_k = \begin{bmatrix}
-c_u(u_k, g_k)^{-1} c_g(u_k, g_k) \\
I
\end{bmatrix},
\]

where \( I : \mathcal{G} \rightarrow \mathcal{G} \) is the identity operator. For SQP algorithms based on this approach, see \([29, 52, 55]\).

**The Augmented System Approach**

A third alternative, related to the approach used by Byrd, et al., in the KNITRO optimization package \([19]\), is a full–space formulation based on the original problem formulation (2.1). Here \( \mathcal{Z} = \mathcal{X} \) and \( W_k \) is the projection onto the null space of \( c_x(x_k) \).

The application \( z = W_k r \) of \( W_k : \mathcal{X} \rightarrow \mathcal{X} \) with \( z, r \in \mathcal{X} \), is computed by solving the augmented system

\[
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
y
\end{pmatrix}
= \begin{pmatrix}
r \\
0
\end{pmatrix},
\]

(2.9)

where \( I : \mathcal{X} \rightarrow \mathcal{X} \) is the identity operator. We should note that the above \( W_k \) is an orthogonal projector, i.e. \( W_k = W_k^* = W_k W_k \). This approach is attractive if efficient iterative solvers for the augmented system (2.9) are available \([4, 8, 14, 15, 53]\).

**2.1.2 Lagrange Multipliers**

The computation of a Lagrange multiplier estimate \( \lambda_k \) at each iterate \( x_k \) is related to the least-squares problem

\[
\min_{\lambda} ||\nabla_x f(x_k) + c_x(x_k)^* \lambda||_\mathcal{X}.
\]

(2.10)

The computation of \( \lambda_k \) is typically related to the choice of the null–space representation \( W_k \).
If the null–space representation $W_k$ is computed using the QR approach, see Section 2.1.1, then the solution of (2.10) can be computed as
\[ \lambda_k = R_k^{-1} Q_k^T \nabla_x f(x_k). \] (2.11)

If instead we use the augmented system approach, then the solution of (2.10) can be computed by solving the linear system
\[
\begin{pmatrix}
I & c_x(x_k) \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
\lambda_k
\end{pmatrix} =
\begin{pmatrix}
-\nabla_x f(x_k) \\
0
\end{pmatrix}.
\] (2.12)

Alternatively, if the basis–nonbasis approach is used, see [52], then the Lagrange multipliers $\lambda_k$ can be estimated as
\[ \lambda_k = -c_u(u_k, g_k)^{-1} \nabla_u f(u_k, g_k). \] (2.13)

In this case $\lambda_k$ does not solve (2.10), but is still an excellent estimate.

### 2.1.3 Hessian Approximations

In addition to the exact Hessian, quasi-Newton approximations can be used to generate $\{H_k\}$. For a detailed discussion of the choices in the SQP context, see [22, p.627–630]. Also see [12].

### 2.1.4 Solution of the Quadratic Subproblem

In this section we describe a composite-step approach for the solution of the quadratic subproblem (2.6). The composite step $s_k^c$ is the combination of a quasi-normal step $n_k$ that aims to (approximately) satisfy the linear constraints, and a tangential step $t_k$ that lies in the tangent space of the constraints and that tries to achieve optimality. A simple illustration of the quasi-normal and tangential steps is given in Figure
We adopt a method similar to that of Byrd and Omojokun [62]. The core ideas of this method have been successfully used and refined by Byrd, Gilbert, and Nocedal [18], Byrd, Hribar, and Nocedal [19], Dennis, El-Alem, and Maciel [28], Dennis, Heinkenschloss, and Vicente [29] [30], and many others.

Figure 2.1: The composite–step approach. For illustrative purposes, it is assumed that $n_k$ is actually normal to the null space of the linearized constraints.

2.1.4.1 Cauchy Decrease Condition

Before we discuss the details of the computation of the quasi-normal and tangential steps, we must turn our attention to a concept crucial to the convergence theory of trust-region methods in general. For this purpose, we consider the unconstrained problem

$$\min f(x) \quad (2.14)$$
where \( x \in \mathcal{X} \), and \( f : \mathcal{X} \to \mathbb{R} \) is a continuously differentiable functional. A trust-region algorithm computes for every iterate \( x_k \) and a given trust-region radius \( \Delta \) a trial step \( s_k \in \mathcal{X} \) as an approximate solution of the problem

\[
\min \quad m_f(x_k, s) \equiv f(x_k) + \langle \nabla_x f(x_k), s \rangle_X + \frac{1}{2} \langle B(x_k) s, s \rangle_X
\]

s.t. \( \|s\| \leq \Delta \),

where \( B(x_k) \) is (or approximates) the Hessian operator \( \nabla_{xx} f(x_k) \).

The step \( s_k \) is required to satisfy what is referred to as a fraction of Cauchy decrease condition to ensure global convergence. What this means is that, for some constant \( \tau > 0 \) independent of \( k \), we require that

\[
m_f(x_k, 0) - m_f(x_k, s_k) \geq \tau (m_f(x_k, 0) - m_f(x_k, s^{cp})) ,
\]

where the Cauchy step \( s^{cp} \) is defined as \( s^{cp} = -\gamma^{cp} \nabla f(x_k) \) with

\[
\gamma^{cp} = \begin{cases} 
\frac{\|\nabla f(x_k)\|_X^2}{\langle B(x_k) \nabla f(x_k), \nabla f(x_k) \rangle_X} & \text{if } \frac{\|\nabla f(x_k)\|_X^2}{\langle B(x_k) \nabla f(x_k), \nabla f(x_k) \rangle_X} \leq \Delta \text{ and } \\
\frac{\Delta}{\|\nabla f(x_k)\|_X} & \langle B(x_k) \nabla f(x_k), \nabla f(x_k) \rangle_X > 0,
\end{cases}
\]

otherwise.

In the latter sections, we will make frequent use of the following lemma, due to Powell [64]. It gives a weaker, yet more compact statement that is used in proofs of convergence of trust-region algorithms instead of the condition (2.15).

**Lemma 2.1.1.** If a step \( s_k \) satisfies a fraction of Cauchy decrease condition, then

\[
m_f(x_k, 0) - m_f(x_k, s_k) \geq \frac{\tau}{2} \|\nabla f(x_k)\|_X \min \left\{ \frac{\|\nabla f(x_k)\|_X}{\|B(x_k)\|_X}, \Delta \right\} .
\]

(2.16)
2.1.4.2 Quasi-Normal Step

The two requirements on the quasi-normal step are, first, that it lie well within the trust region, and second, satisfy the constraint \((2.6b)\) in some sense. For example, we choose a fixed parameter \(\zeta \in (0, 1)\) and ask that the quasi-normal step \(n_k\) approximately solve a problem of the following type:

\[
\begin{align}
\min & \quad \|c_x(x_k)n + c(x_k)\|^2_Y \\
\text{s.t.} & \quad \|n\|_X \leq \zeta \Delta_k.
\end{align}
\]

Global convergence theory for composite-step SQP algorithms does not require that the quasi-normal step exactly satisfy \((2.17)\). In fact, \(n\) need not even be normal to the tangent space (thus the designation “quasi-normal”). Instead, we usually ask that it satisfy a fraction of Cauchy decrease condition on the quadratic model \(m(x_k, n)\) of the linearized constraints,

\[
m(x_k, n) = \|c_x(x_k)n + c(x_k)\|^2_Y.
\]

In particular, \((2.15)\) applied to the model \((2.18)\) suggests that we require from \(n_k\) to give as much decrease as \(n_k^{cp} = -\alpha^{cp}c_x(x_k)^*c(x_k)\), where the step length \(\alpha^{cp}\) is given by

\[
\alpha^{cp} = \begin{cases} 
\frac{\|c_x(x_k)^*c(x_k)\|^2_X}{\|c_x(x_k)c_x(x_k)^*c(x_k)\|^2_Y} & \text{if } \frac{\|c_x(x_k)^*c(x_k)\|^2_X}{\|c_x(x_k)c_x(x_k)^*c(x_k)\|^2_Y} \leq \zeta \Delta_k \\
\frac{\zeta \Delta_k}{\|c_x(x_k)^*c(x_k)\|_X} & \text{otherwise}
\end{cases}
\]

and

\[
m(x_k, 0) - m(x_k, n_k) \geq \sigma_1 \left(\|c(x_k)\|^2_Y - \|c_x(x_k)n_k^{cp} + c(x_k)\|^2_Y\right),
\]

for some \(0 < \sigma_1 \leq 1\) independent of \(k\). Additionally, due to possible nonnormality of \(n_k\) with respect to the tangent space of the constraints, the following condition must
hold for $\kappa_1 > 0$ independent of $k$

$$\|n_k\|_X \leq \kappa_1\|c(x_k)\|_Y.$$  \hfill (2.21)

Requirement (2.20) implies a more compact condition used by Dennis, El-Alem, and Maciel in [28] and Dennis, El-Alem, and Maciel in [28], stated in terms of the following lemma.

**Lemma 2.1.2.** If the quasi-normal step $n_k$ satisfies a fraction of Cauchy decrease condition associated with the trust-region problem (2.17), then

$$m(x_k, 0) - m(x_k, n_k) \geq \kappa_2\|c(x_k)\|_Y \min \{\kappa_3\|c(x_k)\|_Y, \Delta_k\},$$  \hfill (2.22)

for some positive constants $\kappa_2$ and $\kappa_3$ independent of $k$.

**Proof.** The proof is a direct consequence of the application of Lemma 2.1.1 to the model (2.18), followed by the use of the boundedness property of $c_x(x)$ in assumption (A4), the norm inequality

$$\|c(x_k)\|_Y = \|(c_x(x_k)c_x(x_k)^*)^{-1}(c_x(x_k)c_x(x_k)^*)c(x_k)\|_Y \leq \|(c_x(x_k)c_x(x_k)^*)^{-1}\|_{L(Y)}\|c_x(x_k)\|_{L(X,Y)}\|c_x(x_k)c_x(x_k)^*c(x_k)\|_X,$$

and the use of the boundedness property of $(c_x(x_k)c_x(x_k)^*)^{-1}$ in assumption (A4). \hfill \Box

Requirement (2.22) replaces the fraction of Cauchy decrease condition in convergence proofs for trust–region SQP methods.

A concrete procedure for the computation of the quasi-normal step is given in Algorithm 2.1.3. We use the popular dogleg method, proposed by Powell, [63]. It is based on the computation of the Cauchy point $n_k^C$ and the so called Newton step $n_k^N$ (more generally referred to as the feasibility step). The computation of $n_k^N$ is typically tied to the choice of the null–space representation $W_k$. 
If the null–space representer \( W_k \) from Section 2.1.1 is computed using the QR approach or the augmented system approach, then the feasibility step can be computed as the minimum norm minimizer of (2.18):

\[
n_N^k = -c_x(x_k)^* (c_x(x_k)c_x(x_k)^*)^{-1} c(x_k).
\]  

(2.23)

If the null–space representer \( W_k \) from Section 2.1.1 is computed using the QR approach, then the QR factorization can be reused to compute the feasibility step. A simple argument shows that expression (2.23) simplifies to

\[
n_N^k = -Q_k^1 P_k^1 -^T c(x_k).
\]  

(2.24)

If the augmented system approach is used instead, the feasibility step (2.23) can be computed by finding the solution \( y \in \mathcal{Y}, z \in \mathcal{X} \) of the linear system

\[
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
y
\end{pmatrix}
=
\begin{pmatrix}
0 \\
c(x_k)
\end{pmatrix}
\]  

(2.25)

and setting

\[
n_N^k = c_x(x_k)^* y.
\]  

(2.26)

If the basis–nonbasis approach is used, then

\[
n_N^k = \begin{pmatrix}
-c_u(u_k, g_k)^{-1} c(u_k, g_k) \\
0
\end{pmatrix}
\]  

(2.27)

is the adequate representation of the feasibility step. In this case \( n_N^k \) is not perpendicular to the null–space of \( c_x(x_k) \).

The precise statement of the dogleg algorithm follows.

**Algorithm 2.1.3.** (Dogleg Method for the Quasi-Normal Subproblem.)

1. Compute \( n_{k,full}^{cp} = \frac{\|c_x(x_k)^* c(x_k)\|_3^2}{\|c_x(x_k)c_x(x_k)^* c(x_k)\|_5^3} c_x(x_k)^* c(x_k) \).
2. If \( \|n_{k,full}^c\|_X > \zeta \Delta_k \), then

Compute \( \theta_k^c \in (0, 1) \) such that \( \|\theta_k^c n_{k,full}^c\|_X = \zeta \Delta_k \).
Set \( n_k = \theta_k^c n_{k,full}^c \).

Else

(a) Compute \( n_k^N \) via (2.24), (2.25) and (2.26), or (2.27).
(b) If \( \|n_k^N\|_X > \zeta \Delta_k \), then

Compute \( \theta_k^N \in (0, 1) \) such that \( \|n_{k,full}^c + \theta_k^N (n_k^N - n_{k,full}^c)\|_X = \zeta \Delta_k \).
Set \( n_k = n_{k,full}^c + \theta_k^N (n_k^N - n_{k,full}^c) \).

Else
Set \( n_k = n_k^N \).
\[ \square \]

2.1.4.3 Tangential Step

The standard requirements on the tangential step are that it lie in the tangent space of the constraints and that it move the quadratic model of the Lagrangian toward optimality. Thus we usually ask that the tangential step \( t_k \) approximately solve a problem of the type

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle H_k(t + n_k), t + n_k \rangle_X + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), t + n_k \rangle_X + \mathcal{L}(x_k, \lambda_k) \quad (2.28a) \\
\text{s.t.} & \quad c_x(x_k) t = 0 \\
& \quad \|t + n_k\|_X \leq \Delta_k. \quad (2.28b)
\end{align*}
\]

Again, theoretical convergence results do not require an accurate solution of (2.28).
To state an analog of the condition (2.1.2) for the tangential step, we first define the quadratic model of the Lagrangian, given by

\[
q_k^X(s) \equiv \mathcal{L}(x_k, \lambda_k) + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s \rangle_X + \frac{1}{2} \langle H_k s, s \rangle_X.
\]

Due to assumption (A5), for every \( t \in X \), \( c_x(x_k) t = 0 \) if and only if there exists a \( w \in Z \) with the property \( W_k w = t \). Thus, problem (2.28) can be reformulated as the
one of finding the solution \( w_k \) of

\[
\min \quad q_k^Z(w) \equiv q_k^X(n_k) + \langle W_k^*(H_kn_k + \nabla_x \mathcal{L}(x_k, \lambda_k), w \rangle_Z + \frac{1}{2} \langle W_k^*H_kW_kw, w \rangle_Z
\]

\begin{align*}
\text{s.t.} \quad & \|n_k + W_kw\|_X \leq \Delta_k, \\
& \|n_k\|_X \leq \zeta \Delta_k, \quad \text{any } w \text{ with } \|W_kw\|_X \leq \Delta_k - \|n_k\|_X \text{ satisfies the trust region constraint in (2.30).}
\end{align*}

(2.30a)

(2.30b)

and setting \( t_k = W_kw \).

Note that since \( \|n_k\|_X \leq \zeta \Delta_k \), any \( w \) with \( \|W_kw\|_X \leq \Delta_k - \|n_k\|_X \) satisfies the trust region constraint in (2.30). The trust region constraints in (2.28), (2.30) can be replaced by \( \|t\|_X \leq \Delta_k - \|n_k\|_X \) and \( \|W_kw\|_X \leq \Delta_k - \|n_k\|_X \), respectively.

As usual, it is not necessary that \( w_k \) solves (2.30) exactly, but we only require that it satisfy a fraction of the Cauchy decrease condition associated with the trust-region problem (2.30). We will skip the details of the derivation of the resulting compact condition, as it closely follows the procedure used in Section 2.1.4.2 and simply state the result.

**Lemma 2.1.4.** If the tangential step \( t_k \) satisfies a fraction of Cauchy decrease condition associated with the trust-region problem (2.30), then

\[
q_k^X(n_k) - q_k^X(n_k + t_k) \geq \kappa_4\|W_k^*\nabla q_k^X(n_k)\|_Z \min \left\{ \kappa_5\|W_k^*\nabla q_k^X(n_k)\|_Z, \kappa_6\Delta_k \right\},
\]

(2.31)

for some positive constants \( \kappa_4, \kappa_5, \) and \( \kappa_6 \) independent of \( k \).

Requirement (2.31) replaces the fraction of Cauchy decrease condition in convergence proofs for trust–region SQP methods.

We now present concrete algorithms for the solution of the tangential step subproblem (2.28). The first is based on the reduced–space reformulation (2.30). It is a conjugate gradient (CG) algorithm that utilizes Steihaug–Toint [71] stopping criteria to account for the possibility of negative curvature in the reduced Hessian \( W_k^*H_kW_k \)
and violations of the trust–region constraint. Unlike the original Steihaug–Toint scheme, for general null–space representation \( W_k \), this particular version does not guarantee monotone increase of the norms of the iterates. Therefore, even though the trust region constraint \( \| W_k w_{i+1} \|_X \leq \Delta_k - \| n_k \|_X \) might be violated at some iterate \( w_{i+1} \), if the algorithm is not terminated, a subsequent iterate could again satisfy this condition. This means that we could be terminating Alg. 2.1.5 earlier than necessary. From the point of view of the global convergence theory, this is acceptable.

Algorithm 2.1.5. (Steihaug–Toint CG method for the solution of (2.30))

0. Let \( w_0 = 0 \in Z \). Let \( g_k = \nabla_x \mathcal{L}(x_k, \lambda_k) + H_k n_k, r_0 = W_k^* g_k, p_0 = r_0 \).

1. For \( i = 0, 1, 2, \ldots \)

(a) If \( \langle p_i, W_k^* H_k W_k p_i \rangle_Z \leq 0 \), compute \( \theta > 0 \) such that \( \| W_k (w_i + \theta p_i) \| = \Delta_k - \| n_k \|_X \) and return \( w_{i+1} = w_i + \theta p_i \).

(b) \( \alpha_i = \langle r_i, r_i \rangle_Z / \langle p_i, W_k^* H_k W_k p_i \rangle_Z \)

(c) \( w_{i+1} = w_i + \alpha_i p_i \)

(d) If \( \| W_k w_{i+1} \|_X \geq \Delta_k - \| n_k \|_X \), compute \( \theta > 0 \) such that \( \| W_k (w_i + \theta p_i) \| = \Delta_k - \| n_k \|_X \) and return \( w_{i+1} = w_i + \theta p_i \).

(e) \( r_{i+1} = r_i - \alpha_i W_k^* H_k W_k p_i \)

(f) \( \beta_i = \langle r_{i+1}, r_{i+1} \rangle_Z / \langle r_i, r_i \rangle_Z \)

(g) \( p_{i+1} = r_{i+1} + \beta_i p_i \)

□

If \( W_k \) is the orthogonal null–space projector, such as those given by augmented system solves, see Section 2.1.1, then previous algorithm can be rewritten following [47]. First, we note that if \( n_k \) is normal to the null space of the linearized constraints, which is for example true for quasi-normal steps produced by Algorithm 2.1.3 with \( n_k^N \) computed via (2.25) and (2.26), then \( \| t + n_k \|_X^2 = \| t \|_X^2 + \| n_k \|_X^2 \) for all \( t \) in the null–space of \( c_x(x_k) \). Consequently, in this case, the trust-region constraint \( \| t + n_k \|_X \leq \Delta_k \) can be written as

\[
\| t \|_X \leq \sqrt{\Delta_k^2 - \| n_k \|_X^2}.
\]
Thus we can formulate the tangential subproblem as

$$\min \frac{1}{2} \langle H_k(t + n_k), t + n_k \rangle_X + \langle \nabla_x L(x_k, \lambda_k), t + n_k \rangle_X + \mathcal{L}(x_k, \lambda_k)$$ (2.32a)

s.t. $$c_x(x_k) t = 0$$ (2.32b)

$$\|t\|_X \leq \sqrt{\Delta_k^2 - \|n_k\|_X^2}.$$ (2.32c)

It is solved using a Steihaug–Toint variant of a projected CG method in the full space [47]. Because of the properties of $W_k$, this algorithm retains the monotonicity property of the norms of the iterates. Further details are given below.

**Algorithm 2.1.6. (Full–Space Projected Steihaug CG Method.)**

1. Compute $g_0 = W_k r_0$, where $W_k$ is an orthogonal null space projector.
   Set $p_0 = -g_0$.

2. For $j = 0, 1, 2, ...$
   (a) If $\langle H_k p_j, p_j \rangle_X \leq 0$, then set $t_k = t_j^i + \theta p_j$ where $\theta > 0$ is such that $\|t_k\|_X = \Delta_k$, and stop.
   (b) Set $\alpha = \langle r_j, g_j \rangle_X / \langle H_k p_j, p_j \rangle_X$.
   (c) Set $t_{j+1}^i = t_j^i + \alpha p_j$.
   (d) If $\|t_{j+1}^i\|_X > \Delta_k$, then set $t_k = t_j^i + \theta p_j$ where $\theta > 0$ is such that $\|t_k\|_X = \Delta_k$, and stop.
   (e) Set $r_{j+1} = r_j + \alpha H_k p_j$.
   (f) Compute $g_{j+1} = W_k r_{j+1}$.
   (g) If $\langle g_{j+1}, r_{j+1} \rangle_X < tol$, then set $t_k = t_j^i$ and stop.
   (h) Set $\beta = \langle r_{j+1}, g_{j+1} \rangle_X / \langle r_j, g_j \rangle_X$.
   (i) Set $p_{j+1} = -g_{j+1} + \beta p_j$.

□

We recall that $W_k = W_k^* = W_k W_k$. The consequence of this property is that Alg. 2.1.6 applied to the full–space problem (2.32) is equivalent to Alg. 2.1.5 applied to the corresponding problem in the reduced space.
2.1.5 Acceptance of the Trial Step and the Update of the
Trust–Region Radius

Global convergence of trust–region SQP methods is often ensured through the use of
a merit function, which helps us determine whether a step is acceptable and whether
the trust-region radius needs to be modified. A crucial property of a merit function
is that its critical points should also be the critical points of the underlying problem. Additionally, it is highly desirable that the critical points of the two problems completely coincide.

A variety of merit functions have been used in SQP algorithms (for detailed reviews
see [22, Ch.15] and [60, Ch.18]) as their performance varies depending on the details of
the particular SQP scheme, features of target applications, etc. We use the augmented
Lagrangian merit function

\[ \phi(x, \lambda; \rho) = f(x) + \langle \lambda, c(x) \rangle_Y + \rho \|c(x)\|^2_Y = L(x, \lambda) + \rho \|c(x)\|^2_Y. \] (2.33)

We use this merit function because the few theoretical results on the treatment of
inexact linear system solves in SQP algorithms are based on this particular choice.
Additionally, the convergence theory presented here is mostly based on [28], where
the authors use this merit function.

Let \( s_k^x = n_k + t_k \) be a trial step with \( n_k \) satisfying (2.22) and (2.21), and \( t_k \)
satisfying (2.31), and let \( \lambda_{k+1} = \lambda_k + \Delta \lambda_k \) be an updated Lagrange multiplier. To
measure the improvement in the merit function \( \phi \), we compare the actual reduction
and the predicted reduction in moving from the current iterate \( x_k \) to the trial iterate
\( x_k + s_k^x \). The actual reduction is defined by

\[ \text{ared}(s_k^x; \rho_k) = \phi(x_k, \lambda_k; \rho_k) - \phi(x_k + s_k^x, \lambda_{k+1}; \rho_k), \] (2.34)
and the predicted reduction is given by

\[
pred(s_k^x; \rho_k) = \phi(x_k, \lambda_k; \rho_k) \\
- \left( q_k^\nu(s_k^x) + \langle \Delta \lambda_k, c_x(x_k)s_k^x + c(x_k) \rangle \right) + \rho_k \|c_x(x_k)s_k^x + c(x_k)\|^2_Y.
\]

(2.35)

Two issues remain to be discussed. The first is the examination of the computed step together with an update strategy for the trust-region radius. It is directly addressed in Algorithm 2.1.8. The second is the update of the penalty parameter \( \rho_k \).

This is a crucial issue for SQP methods that employ the augmented Lagrangian merit function. We use El-Alem’s scheme given in [35], and adopted by Dennis, El-Alem, and Maciel in [28], and Heinkenschloss and Vicente in [55]. The scheme guarantees a decrease in the merit function proportional to a fraction of Cauchy decrease in the quadratic model (2.18) of the constraints. It is stated in Algorithm 2.1.7 which is essentially the same as the penalty update method in [28]. An important observation is that a penalty parameter \( \rho_k \) generated by this algorithm always satisfies

\[
pred(s_k^x; \rho_k) \geq \frac{\rho_k}{2} \left( \|c(x_k)\|_Y^2 - \|c_x(x_k)s_k^x + c(x_k)\|_Y^2 \right).
\]

(2.36)

Algorithm 2.1.7. (Penalty Parameter Update.)

1. Assumption: Let \( \rho_{-1} = 1 \) and let \( \beta \) be a small positive constant.

2. If

\[
pred(s_k^x; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2} \left( \|c(x_k)\|_Y^2 - \|c_x(x_k)s_k^x + c(x_k)\|_Y^2 \right)
\]

then set \( \rho_k = \rho_{k-1} \). Otherwise set

\[
\rho_k = \frac{2 \left( q_k^\nu(s_k^x) - q_k^\nu(0) + \langle \Delta \lambda_k, c_x(x_k)s_k^x + c(x_k) \rangle \right) + \beta}{\|c(x_k)\|_Y^2 - \|c_x(x_k)s_k^x + c(x_k)\|_Y^2}.
\]

□

We now have the necessary pieces to formulate the version of our trust-region SQP algorithm based on exact linear system solves, see Algorithm 2.1.8.

Algorithm 2.1.8. (Trust-region SQP algorithm.)
1. Initialization. Choose initial point $x_0$, initial trust-region radius $\Delta_0$, and constants $0 < \eta_1 < \eta_2 < 1$, $0 < \gamma_1 < \gamma_2 < 1$, $\rho - 1 \geq 1$, and $\epsilon_{tol} > 0$. Compute an initial Lagrange multiplier estimate $\lambda_0$.

2. For $k = 0, 1, 2, \ldots$

   (a) Convergence Check. If
   \[
   \| W_k^* \nabla_x \mathcal{L}(x_k, \lambda_k) \|_Z + \| C(x_k) \|_Y < \epsilon_{tol},
   \]
   then terminate.

   (b) Trial Step Computation.
   
   i. Find a quasi-normal step $n_k$ satisfying (2.22) and (2.21).
   
   ii. Find a tangential step $t_k$ satisfying (2.31).
   
   iii. Set $s_k = n_k + t_k$.
   
   iv. Compute a new Lagrange multiplier estimate $\lambda_{k+1}$.

   (c) Acceptance Test.
   
   i. Update penalty parameter $\rho_k$ via Algorithm 2.1.7.
   
   ii. Compute actual reduction $\text{ared}(s_k^\tau; \rho_k)$ and predicted reduction $\text{pred}(s_k^\tau; \rho_k)$, and their ratio $\theta_k = \frac{\text{ared}(s_k^\tau; \rho_k)}{\text{pred}(s_k^\tau; \rho_k)}$.
   
   iii. If $\theta_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$, otherwise set $x_{k+1} = x_k$ and reset $\lambda_{k+1} = \lambda_k$.

   (d) Trust-Region Radius Update. Set
   \[
   \Delta_{k+1} \in \begin{cases} 
   [\Delta_k, \infty] & \text{if } \theta_k \geq \eta_2, \\
   [\gamma_2 \Delta_k, \Delta_k] & \text{if } \theta_k \in [\eta_1, \eta_2), \\
   [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \theta_k < \eta_1.
   \end{cases}
   \]

☐

We also state a basic convergence result. For a detailed proof, see [28].

**Theorem 2.1.9.** Under the general problem assumptions (A1)-(A7), the sequence of iterates generated by Algorithm 2.1.8 satisfies
\[
\liminf_{k \to \infty} (\| W_k^* \nabla_x \mathcal{L}(x_k, \lambda_k) \|_Z + \| C(x_k) \|_Y) = 0.
\]
Moreover, given any $\epsilon_{tol} > 0$, the algorithm terminates in finitely many iterations, due to

$$\|W^*_k \nabla_x \mathcal{L}(x_k, \lambda_k)\|_Z + \|C(x_k)\|_Y < \epsilon_{tol}.$$ 

This thesis is primarily concerned with the global convergence to a point satisfying the first order necessary optimality conditions. Global convergence of algorithms of the type 2.1.8 to a point that satisfies the second order necessary optimality conditions is shown in [30].

Finally, we remind the reader that the trust-region approach is only one of the possible globalization schemes for SQP algorithms. Line search techniques have been studied and effectively applied in this context, see, for example, [68] [73]. Additionally, in the late 1990s the concept of filter methods emerged in the study of SQP techniques, see Fletcher, et al. [36] [37].
Chapter 3

Inexactness in a Generalized Trust–Region SQP Algorithm

As presented in Chapter 2, the computation of Lagrange multipliers, the quasi–normal step, and the tangential step each involve (potentially multiple) linear system solves. For large–scale problems utilizing augmented system or basis–nonbasis null space representations in the tangential step algorithm, with corresponding Lagrange multiplier and quasi–normal step computations (see Section 2.1.1), all linear systems are solved iteratively. In this case, one can usually only guarantee that their approximate solutions give linear system residuals that are below a specified tolerance. Therefore, in this scenario, the quasi-normal step $n_k$, the Lagrange multipliers $\lambda_k$, and the tangential step $t_k$ are never computed exactly. The main question here is how to ensure global convergence of the optimization algorithm by controlling the size of the linear system residuals in the computation of the Lagrange multipliers, the quasi–normal step, and the tangential step. Additionally, the interaction between the inexactness in the quasi–normal and tangential step must be closely examined. One important issue is that the inexactly computed tangential step can leave the null space of the linearized constraints, see Figure 3.1, which can affect the amount of linear feasibility gained in the quasi-normal step computation, as well as the criteria for the acceptance of the composite step (penalty parameter update, definition of predicted reduction).
Global convergence theory of trust–region methods for unconstrained optimization \[21, 48\] and trust–region SQP methods \[28, 48\] provides important guidelines for the treatment of the issue of inexactness in composite–step trust–region SQP algorithms. The work of Heinkenschloss and Vicente \[55\] builds on the generic trust–region SQP convergence theory and establishes a theoretical framework for the treatment of inexactness arising in iterative linear system solves. The mechanisms of inexactness control developed in \[55\] do not depend on Lipschitz constants and derivative norm bounds, as is the case with other attempts in this area (see \[59\]), and are therefore computationally attractive. However, the authors do not fully analyze the practical implications of suggested theoretical mechanisms. Also, their work is limited to the reduced–space SQP approach, which assumes the basis–nonbasis partitioning of optimization variables.

The two goals of our work are to extend the theoretical inexactness framework of Heinkenschloss and Vicente to trust–region SQP methods for general equality–
constrained optimization problems, and to devise a practical inexact SQP framework, i.e. a concrete optimization algorithm that efficiently manages the stopping criteria for iterative linear system solves. This chapter establishes the theoretical basis for a generalized inexactness framework. Chapter 4 specializes this generic framework to the case in which the null-space representer $W_k$ is an orthogonal null-space projection.

Section 3.1 states the convergence conditions that must be satisfied by the quasi-normal step. Section 3.2 addresses the global convergence conditions for the tangential step. In Section 3.3, we address the issue of balancing progress in feasibility and optimality in presence of inexactness. A generalized SQP algorithm based on our theoretical framework for inexactness control is stated in Section 3.4. In Section 3.5, we prove that this algorithm is globally convergent. Section 3.6 focuses on the issue of inexact linear system solves in the tangential step computation. The results of this section are specialized to the case in which $W_k$ is a projector in Chapter 4.

### 3.1 Inexactness in the Quasi–Normal Step

For completeness, in this section we merely restate the global convergence requirements that must be imposed on the quasi–normal step, in the inexact setting. We require the boundedness condition

$$
\|n_k\|_X \leq \kappa_1 \|c(x_k)\|_Y,
$$

(3.1)

where $\kappa_1 > 0$ is independent of $k$, and the fraction of Cauchy decrease condition

$$
\|c(x_k)\|_Y^2 - \|c_x(x_k)n_k + c(x_k)\|_Y^2 \geq \kappa_2 \|c(x_k)\|_Y \min \left\{\kappa_3 \|c(x_k)\|_Y, \Delta_k\right\},
$$

(3.2)

where $\kappa_2, \kappa_3 > 0$ are independent of $k$. 
3.2 Inexactness in the Tangential Step

The tangential subproblem can be formulated as

\[
\begin{align*}
& \min \frac{1}{2} \langle H_k t, t \rangle_X + \langle g_k, t \rangle_X + q_k^\chi(n_k) \\
& \text{s.t.} \quad c_x(x_k)t = 0 \\
& \quad \|t\|_X \leq \Delta_k - \|n_k\|_X,
\end{align*}
\]

where

\[ g_k = H_k n_k + \nabla_x \mathcal{L}(x_k, \lambda_k). \]

We now recall that we have previously introduced the (exact) null-space operator \( W_k : \mathcal{Z} \to \mathcal{X} \) that satisfies \( \text{Range}(W_k) = \text{Null}(c_x(x_k)) \). This operator is not a mere theoretical construct. Its purpose is to cover all existing practical implementations for handling the linear constraint \( c_x(x_k)t = 0 \), and at the same time generalize the analysis. We let \( t_k = W_k w_k \), and problem (3.3) becomes

\[
\begin{align*}
& \min \frac{1}{2} \langle W_k^* H_k W_k w, w \rangle_{\mathcal{Z}} + \langle W_k^* g_k, w \rangle_{\mathcal{Z}} + q_k^\chi(n_k) \\
& \text{s.t.} \quad \|W_k w\|_X \leq \Delta_k - \|n_k\|_X.
\end{align*}
\]

We call (3.4) the reduced tangential subproblem.

With inexact linear system solves in the application of \( W_k \) and \( W_k^* \), the reduced gradient \( W_k^* g_k \) and the reduced Hessian \( W_k^* H_k W_k \) are no longer available. We hope to replace the exact null-space operator \( W_k \) with an inexact representation \( \widetilde{W}_k \), and introduce the inexact reduced gradient vector \( \widetilde{W}_k^* g_k \) and the inexact reduced Hessian operator \( \widetilde{W}_k^* H_k \widetilde{W}_k \). This allows us to impose convergence criteria arising in the treatment of inexact trust-region methods for unconstrained optimization.

Under the assumption that we can introduce a fixed inexact reduced gradient and a fixed inexact reduced Hessian in the tangential subproblem (3.4), we obtain its
inexact counterpart

\[
\min \quad \tilde{q}_k(w) \equiv \frac{1}{2} \left\langle \tilde{W}_k^s H_k W_k w, w \right\rangle_Z + \left\langle \tilde{W}_k g_k, w \right\rangle_Z + q^X_k(n_k) \quad (3.5a)
\]

\[
\text{s.t.} \quad \|\tilde{W}_k w\|_X \leq \Delta_k - \|n_k\|_X. \quad (3.5b)
\]

The theoretical work on the global convergence of trust-region algorithms using inexact gradient and Hessian information, in the context of unconstrained optimization [21, 22], allows us to pose convergence requirements on \(\tilde{W}_k g_k\) and \(\tilde{W}_k^s H_k W_k\). The inexact reduced gradient needs to satisfy

\[
\|\tilde{W}_k^s g_k - W_k^s g_k\|_Z \leq \xi_1 \min \left( \|\tilde{W}_k^s g_k\|_Z, \Delta_k \right),
\]

for some \(\xi_1 > 0\) independent of \(k\). The inexact reduced Hessian must satisfy

\[
\left\langle \tilde{W}_k^s H_k W_k w_k, w_k \right\rangle_Z \leq \xi_2 \|w_k\|_Z^2,
\]

for some \(\xi_2 > 0\) independent of \(k\). Finally, we also need to impose the fraction of Cauchy decrease condition on \(w_k\) with respect to the inexact quadratic model \(\tilde{q}_k\),

\[
\tilde{q}_k(0) - \tilde{q}_k(w_k) \geq \kappa_4 \|\tilde{W}_k^s g_k\|_Z \min \left\{ \kappa_5 \|\tilde{W}_k^s g_k\|_Z, \kappa_6 \Delta_k \right\},
\]

for \(\kappa_4, \kappa_5, \kappa_6 > 0\), independent of \(k\). Depending on the algorithm for the solution of the reduced quadratic subproblem, establishing the existence of \(\tilde{W}_k\), \(\tilde{W}_k^s g_k\), and \(\tilde{W}_k^s H_k W_k\) is by no means trivial, and is a subject of study of Section 3.6.
3.3 Balancing Progress in the Computation of the Tangential and the Quasi–Normal Step

Assuming that we have computed the reduced-space step $w_k \in \mathcal{Z}$ satisfying the requirements discussed in Section 3.2, we still have to perform the computation of the tangential step $\tilde{t}_k \in \mathcal{X}$. With exact linear system solves, $\tilde{t}_k = W_k w_k$. Since the application of $W_k$ requires the solution of linear systems, it is not possible to compute $W_k w_k$ exactly. Hence, we need to determine how much the effective tangential step $\tilde{t}_k \in \mathcal{X}$ can differ from $W_k w_k$.

In order to better understand how to balance feasibility and optimality in presence of inexactness, we revisit the step acceptance criteria from Section 2.1.5, in particular the measure of predicted reduction. In exact arithmetic, the following holds

$$c_x(x_k)(n_k + W_k w_k) = c_x(x_k)n_k,$$

thus we obtain for the predicted reduction defined in (2.35)

$$\text{pred}(s_k^x; \rho_k) = \phi(x_k, \lambda_k; \rho_k) - \left[ \mathcal{L}(x_k, \lambda_k) + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s_k^x \rangle_{\mathcal{X}} + \frac{1}{2} \langle H_k s_k^x, s_k^x \rangle_{\mathcal{X}} \right]
+ \langle \lambda_{k+1} - \lambda_k, c_x(x_k)s_k^x + c(x_k) \rangle_{\mathcal{Y}} + \rho_k \| c_x(x_k)s_k^x + c(x_k) \|^2_{\mathcal{Y}}
= \mathcal{L}(x_k, \lambda_k) - \left( \mathcal{L}(x_k, \lambda_k) + \langle W_k^* (\nabla_x \mathcal{L}(x_k, \lambda_k) + H_k n_k), w_k \rangle_{\mathcal{Z}}
+ \frac{1}{2} \langle W_k^* H_k W_k w_k, w_k \rangle_{\mathcal{Z}} \right)
- \langle \nabla_x \mathcal{L}(x_k, \lambda_k), n_k \rangle_{\mathcal{X}} - \frac{1}{2} \langle H_k n_k, n_k \rangle_{\mathcal{X}}
- \langle \lambda_{k+1} - \lambda_k, c_x(x_k)n_k + c(x_k) \rangle_{\mathcal{Y}}
+ \rho_k \left( \| c(x_k) \|^2_{\mathcal{Y}} - \| c_x(x_k)n_k + c(x_k) \|^2_{\mathcal{Y}} \right).$$
Therefore,

\[
\text{pred}(s_k^x; \rho_k) = - \langle W^*_k (\nabla_x L(x_k, \lambda_k) + H_k n_k), w_k \rangle_Z - \frac{1}{2} \langle W^*_k H_k W_k w_k, w_k \rangle_Z \\
- \langle \nabla_x L(x_k, \lambda_k), n_k \rangle_X - \frac{1}{2} \langle H_k n_k, n_k \rangle_X \\
- \langle \lambda_{k+1} - \lambda_k, c_x(x_k)n_k + c(x_k) \rangle_Y \\
+ \rho_k \left( \|c(x_k)\|^2_3 - \|c_x(x_k)n_k + c(x_k)\|^2_Y \right) .
\]

With inexact linear system solves the trial step is \( s_k^x = n_k + \tilde{t}_k \), with

\[
c_x(x_k)(n_k + \tilde{t}_k) = c_x(x_k)n_k + r_k^t \neq c_x(x_k)n_k,
\]

where we have defined

\[
r_k^t \equiv c_x(x_k)\tilde{t}_k.
\]

This will effect the term

\[
- \langle \lambda_{k+1} - \lambda_k, c_x(x_k)n_k + c(x_k) \rangle_Y + \rho_k \left( \|c(x_k)\|^2_3 - \|c_x(x_k)n_k + c(x_k)\|^2_Y \right)
\]

in the representation of \( \text{pred}(s_k^x; \rho_k) \) given above.

For the predicted reduction with inexactness we replace

\[
- \langle W^*_k (\nabla_x L(x_k, \lambda_k) + H_k n_k), w_k \rangle_Z - \frac{1}{2} \langle W^*_k H_k W_k w_k, w_k \rangle_Z
\]

by

\[
- \langle \tilde{W}_k^* g_k, w_k \rangle_Z - \frac{1}{2} \langle \tilde{W}_k^* H_k W_k w_k, w_k \rangle_Z = \tilde{q}_k(0) - \tilde{q}_k(w_k),
\]

and replace \( n_k \) by \( n_k + \tilde{t}_k \) in

\[
- \langle \lambda_{k+1} - \lambda_k, c_x(x_k)n_k + c(x_k) \rangle_Y + \rho_k \left( \|c(x_k)\|^2_3 - \|c_x(x_k)n_k + c(x_k)\|^2_Y \right) .
\]
We consequently obtain for the predicted reduction with inexactness

\[
\tilde{\text{pred}}(s_k^*; \rho_k) \equiv \text{pred}(n_k, w_k; \rho_k) + r\text{pred}(r_k^t; \rho_k),
\]

where

\[
\text{pred}(n_k, w_k; \rho_k) \equiv \tilde{q}_k(0) - \tilde{q}_k(w_k) - \langle \nabla_x \mathcal{L}(x_k, \lambda_k), n_k \rangle_x - \frac{1}{2} \langle H_k n_k, n_k \rangle_x
\]

\[
+ \langle \lambda_{k+1} - \lambda_k, c_x(x_k) n_k + c(x_k) \rangle_y
\]

\[
+ \rho_k \left( \|c(x_k)\|_Y^2 - \|c_x(x_k) n_k + c(x_k)\|_Y^2 \right),
\]

and

\[
r\text{pred}(r_k^t; \rho_k) \equiv - \langle \lambda_{k+1} - \lambda_k, r_k^t \rangle_y - \rho_k \|r_k^t\|_Y^2 - 2\rho_k \langle r_k^t, c_x(x_k) n_k + c(x_k) \rangle_y.
\]

In practice, we first compute a penalty parameter \(\rho_k\) satisfying

\[
\text{pred}(n_k, w_k; \rho_k) \geq \frac{\rho_k}{2} \left( \|c(x_k)\|_Y^2 - \|c_x(x_k) n_k + c(x_k)\|_Y^2 \right),
\]

and then make sure that the effective tangential step \(\tilde{t}_k\) satisfies the requirement

\[
|r\text{pred}(r_k^t; \rho_k)| \leq \eta_0 \text{pred}(n_k, w_k; \rho_k),
\]

where \(\eta_0 \in (0, 1 - \eta_1)\), and \(\eta_1 \in (0, 1)\) is the smallest acceptable ratio of the actual and predicted reduction.

There are three additional, rather technical conditions that need to be satisfied by the effective tangential step. They are required for the proof of global convergence of Alg. 3.4.2. The first is related to how much the effective tangential step can deviate from \(W_k w_k\), and reads

\[
\|\tilde{t}_k - W_k w_k\|_X \leq \xi_3 \Delta_k \|s_k^\ast\|_X,
\]

(3.12)
for some $\xi_3 > 0$ independent of $k$. The second is a boundedness condition,

$$\|\tilde{r}_k\|_Y \leq \xi_4 \Delta_k, \quad (3.13)$$

where $\xi_4 > 0$ is independent of $k$. The third condition is only indirectly related to the computation of the effective tangential step. The reduced–space tangential step $w_k$ must satisfy

$$\|w_k\|_Z \leq \xi_5 \|s_k\|_X, \quad (3.14)$$

for $\xi_5 > 0$ independent of $k$.

### 3.4 Statement of the Generalized Trust–Region SQP Algorithm with Inexactness Control

In this section we state a trust–region SQP algorithm that provides a generic theoretical framework for inexactness control.

For the update of the penalty parameter $\rho_k$, we use Alg. 3.4.1.

**Algorithm 3.4.1. (Penalty Parameter Update with Inexactness.)**

1. Assumption: Let $\rho_{-1} = 1$ and let $\bar{\rho}$ be a small positive constant.

2. If

$$\text{pred}(n_k, w_k; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2} \left( \|c(x_k)\|_Y^2 - \|c_x(x_k)n_k + c(x_k)\|_Y^2 \right)$$

then set $\rho_k = \rho_{k-1}$. Otherwise set

$$\rho_k = \frac{2\text{pred}(n_k, w_k; \rho_{k-1})}{\|c(x_k)\|_Y^2 - \|c_x(x_k)n_k + c(x_k)\|_Y^2} + \bar{\rho}.$$  

□

We now have all ingredients necessary to formulate the SQP algorithm in the inexact setting.

**Algorithm 3.4.2. (Trust-region SQP algorithm with inexact linear system solves.)**
1. Initialization. Choose initial point $x_0$, initial trust-region radius $\Delta_0$, constants $0 < \alpha_1, \eta_1 < 1$, $0 < \eta_0 < 1 - \eta_1$, $\rho - 1 \geq 1$, and $\text{tol}^{SQP} > 0$. Set $\Delta_{\text{min}}, \Delta_{\text{max}}$ so that $0 < \Delta_{\text{min}} < \Delta_{\text{max}}$. Compute an initial Lagrange multiplier estimate $\lambda_0$.

2. For $k = 0, 1, 2, \ldots$
   
   (a) Convergence Check. If
   $$\|\nabla_x L(x_k, \lambda_k)\|_X < \text{tol}^{SQP} \quad \text{and} \quad \|c(x_k)\|_Y < \text{tol}^{SQP},$$
   then terminate.

   (b) Step Computation.
   
   i. Compute a quasi-normal step $n_k$ satisfying (3.1) and (3.2).
   
   ii. Compute a reduced-space tangential step $w_k$ satisfying (3.6)--(3.8).

   (c) Acceptance Test.
   
   i. Compute a new Lagrange multiplier estimate $\lambda_{k+1}$ based on $n_k + t_k$.
   
   ii. Update penalty parameter $\rho_k$ via Alg. 3.4.1
   
   iii. Compute effective tangential step $\tilde{t}_k$ satisfying the conditions (3.11), (3.12), (3.13), and (3.14).
   
   iv. Compute trial step $s_k^x = n_k + \tilde{t}_k$.
   
   v. Compute actual reduction $\text{ared}(s_k^x; \rho_k)$ and predicted reduction $\text{pred}(n_k, w_k; \rho_k)$, and their ratio $\theta_k = \frac{\text{ared}(s_k^x; \rho_k)}{\text{pred}(n_k, w_k; \rho_k)}$.
   
   vi. If $\theta_k \geq \eta_1$, set $x_{k+1} = x_k + s_k^x$, and choose $\Delta_{k+1}$ such that
   $$\max\{\Delta_{\text{min}}, \Delta_k\} \leq \Delta_{k+1} \leq \Delta_{\text{max}}.$$ 
   Otherwise set $x_{k+1} = x_k$, reset $\lambda_{k+1} = \lambda_k$, and set
   $$\Delta_{k+1} = \alpha_1 \max\{\|n_k\|_X, \|\tilde{t}_k\|_X\}.$$ 

□

3.5 Global Convergence of Alg. 3.4.2

The global convergence property of Alg. 3.4.2 is stated in the following theorem.

Theorem 3.5.1. Let assumptions (A4)–(A7) be satisfied. The sequences of iterates generated by Alg. 3.4.2 satisfy

$$\liminf_{k \to \infty}(\|W_k g_k\|_Z + \|c(x_k)\|_Y) = 0.$$  (3.15)
Additionally, we have

\[
\lim_{k \to \infty} \inf \left( \|W_k^* \nabla f(x_k)\|_Z + \|c(x_k)\|_Y \right) = 0.
\]  

(3.16)

**Proof.** The proof of (3.15) is based on the global convergence analysis given in [28] and closely follows the modification given in [55]. The component ‘pred\((s_k^\circ, (s_u)_k; \rho_k)\)’ of the predicted decrease in [55] will be replaced by \(pred(n_k, w_k; \rho_k)\) in our notation.

As in [55], the first modification relates the size of the trial step \(s_k^\circ\) to the trust–
region radius \(\Delta_k\). We need that

\[
\|s_k^\circ\|_X \leq \kappa_7 \Delta_k,
\]

and, if \(s_k^\circ\) is rejected,

\[
\|s_k^\circ\|_X \geq \kappa_8 \Delta_k.
\]

The first inequality follows from \(s_k^\circ = n_k + \tilde{t}_k\), the condition (2.17b) (which is always satisfied directly by the corresponding quasi–normal step algorithm), and the requirement (3.13). The second inequality follows trivially from the trust region update in Step 2(c)vi of Alg. 3.4.2.

The second modification concerns the estimates of the actual and the predicted reduction. By analogy to [55] we will be able to show

\[
|\text{ared}(s_k^\circ; \rho_k) - \text{pred}(n_k, w_k; \rho_k) - r\text{pred}(r_k^t; \rho_k)|
\leq \kappa_9 \Delta_k \|s_k^\circ\|_X + \kappa_{10} \rho_k \|s_k^\circ\|^3_X + \kappa_{11} \rho_k \|s_k^\circ\|^2_X \|c(x_k)\|
\]

(3.17)

instead of [28, Lemma 7.4] and

\[
|\text{ared}(s_k^\circ; \rho_k) - \text{pred}(n_k, w_k; \rho_k) - r\text{pred}(r_k^t; \rho_k)| \leq \kappa_{12} \rho_k \Delta_k \|s_k\|_X
\]

(3.18)
instead of \[28\] Lemma 7.5. In the analysis below, we will use the following notation
\[
q^X_k(s) = \mathcal{L}(x_k, \lambda_k) + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s \rangle_{\mathcal{X}} + \frac{1}{2} \langle H_k s, s \rangle_{\mathcal{X}},
\]
\[
q^Z_k(w) = q^X_k(n_k) + \langle W^*_k (H_k n_k + \nabla_x \mathcal{L}(x_k, \lambda_k)), w \rangle_{\mathcal{Z}} + \frac{1}{2} \langle W^*_k H_k W_k w, w \rangle_{\mathcal{Z}},
\]
\[
x_{k+1} = x_k + s^\tau_k,
\]
and recall the definition of the actual reduction
\[
ared(s^\tau_k; \rho_k) = \mathcal{L}(x_k, \lambda_k) + \rho_k ||c(x_k)||^2_Y - \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \rho_k ||c(x_{k+1})||^2_Y.
\]

With definitions \((3.9)\) and \((3.10)\), we obtain
\[
pred(n_k, w_k; \rho_k) + rpred(r^i_k; \rho_k)
\]
\[
= -\langle \hat{W}^*_k g_k, w_k \rangle_{\mathcal{Z}} - \frac{1}{2} \langle \hat{W}^*_k H_k W_k w_k, w_k \rangle_{\mathcal{Z}} - \langle \nabla_x \mathcal{L}(x_k, \lambda_k), n_k \rangle_{\mathcal{X}} - \frac{1}{2} \langle H_k n_k, n_k \rangle_{\mathcal{X}}
\]
\[
- \langle \Delta \lambda_k, c_x(x_k) s^\tau_k + c(x_k) \rangle_Y + \rho_k (||c(x_k)||^2_Y - ||c_x(x_k) s^\tau_k + c(x_k)||^2_Y).
\]

This expression, the fact \(W^*_k (H_k n_k + \nabla_x \mathcal{L}(x_k, \lambda_k)) = W^*_k g_k\), and additional manipulations (see \([55]\)) yield
\[
ared(s^\tau_k; \rho_k) - pred(n_k, w_k; \rho_k) - rpred(r^i_k; \rho_k)
\]
\[
= -\mathcal{L}(x_{k+1}, \lambda_k) + q^X_k(s^\tau_k) - q^X_k(s^\tau_k) + q^Z_k(w_k)
\]
\[
+ \langle \hat{W}^*_k g_k - W^*_k g_k, w_k \rangle_{\mathcal{Z}} + \frac{1}{2} \langle \hat{W}^*_k H_k W_k w_k, w_k \rangle_{\mathcal{Z}} - \frac{1}{2} \langle W^*_k H_k W_k w_k, w_k \rangle_{\mathcal{Z}}
\]
\[
+ \langle \Delta \lambda_k, -c(x_{k+1}) + c_x(x_k) s^\tau_k + c(x_k) \rangle_Y - \rho_k (||c(x_{k+1})||^2_Y - ||c_x(x_k) s^\tau_k + c(x_k)||^2_Y).
\]
\[
(3.19)
\]

A Taylor expansion of \(\mathcal{L}(x_{k+1}, \lambda_k)\) gives
\[
| - \mathcal{L}(x_{k+1}, \lambda_k) + q^X_k(s^\tau_k) | \leq \frac{1}{2} \|H_k - \nabla_{xx} \mathcal{L}(x_k + t^1_k s^\tau_k, \lambda_k)\|_{\mathcal{L}(x_k)} ||s^\tau_k||_{\mathcal{X}},
\]
\[
(3.20)
\]
where $0 < \tau_k^1 < 1$. Definitions of $q_k^\chi$ and $q_k^\xi$ and norm inequalities give

$$| - q_k^\chi(s_k^x) + q_k^\xi(w_k)| \leq \| H_k n_k + \nabla_x \mathcal{L}(x_k, \lambda_k) \|_x \| \tilde{t}_k - W_k w_k \|_x$$

$$+ \frac{1}{2} \| H_k \|_{L(X)} \| \tilde{t}_k \|_X^2 + \frac{1}{2} \| W_k^* H_k W_k \|_{L(Z)} \| w_k \|_Z^2,$$

and so from (3.12), (3.14), and

$$\| \tilde{t}_k \|_X \leq \| \tilde{t}_k - W_k w_k \|_X + \| W_k w_k \|_X,$$

we obtain

$$| - q_k^\chi(s_k^x) + q_k^\xi(w_k)| \leq C_5 \| H_k n_k + \nabla_x \mathcal{L}(x_k, \lambda_k) \|_x \Delta_k \| s_k^x \|_X$$

$$+ \frac{1}{2} \| H_k \|_{L(X)} (\xi_2^2 \Delta_k^2 + 2\xi_3 \xi_5 \| W_k \|_{L(Z)} \| s_k^x \|_{X} \Delta_k + \xi_6 \| W_k \|_{L(Z)}^2 \| s_k^x \|_{X}^2) \| s_k^x \|_X^2$$

$$+ \frac{1}{2} \xi_7 \| W_k^* H_k W_k \|_{L(Z)} \| s_k^x \|_X^2.$$

Inequalities (3.6) and (3.7) yield

$$\langle \tilde{W}_k g_k - W_k^* g_k, w_k \rangle_Z + \frac{1}{2} \langle W_k^* H_k W_k w_k, w_k \rangle_Z - \frac{1}{2} \langle W_k^* H_k W_k w_k, w_k \rangle_Z$$

$$\leq \xi_1 \Delta_k \| s_k^x \|_x + \frac{1}{2} \xi_2 (\| W_k^* H_k W_k \|_{L(Z)} \| w_k \|_Z^2$$

$$\quad \leq \xi_1 \xi_5 \Delta_k \| s_k^x \|_x + \frac{1}{2} \xi_7 \xi_5^2 (\| W_k^* H_k W_k \|_{L(Z)} \| s_k^x \|_X^2 \| s_k^x \|_X^2.$$

The final step follows [55] directly. After a series of Taylor expansions, we obtain

$$\langle \Delta \lambda_k, -c(x_{k+1}) + c(x_k) s_k^x + c(x_k) \rangle_Y - \rho_k (\| c(x_{k+1}) \|_Y^3 - \| c(x_k) s_k^x + c(x_k) \|_Y^3)$$

$$\quad \leq \kappa_{10} \rho_k \| s_k^x \|_X^3 + \kappa_{11} \rho_k \| s_k^x \|_X \| c(x_k) \|_Y.$$

Substitution of (3.20)–(3.23) into (3.19) along with the assumptions (A4)–(A7) gives the estimate (3.17). As $\rho_k \geq 1$, (3.18) follows. The estimates of the difference between
the actual and the predicted reduction can be found in [55], which concludes the part of the proof related to (3.15). The limit (3.16) is obtained by combining (3.15), (3.6), and (3.1).

3.6 A Generalized Conjugate Gradient Method for the Solution of the Tangential Subproblem

The reduced tangential subproblem (3.4) is commonly solved using a conjugate gradient (CG) algorithm with Steihaug–Toint stopping conditions. This is Alg. 2.1.5, which we state here again with slightly different notation. This will be the starting point in the analysis of inexactness.

Algorithm 3.6.1. (Steihaug–Toint CG method for the solution of (3.4))

0. Given tol > 0. Let \( w_0 = 0 \in \mathbb{Z} \). Let \( \tilde{r}_0 = W_k^* g_k, \tilde{p}_0 = \tilde{r}_0 \).

1. For \( i = 0, 1, 2, \ldots \)

   (a) If \( \| \tilde{r}_i \|_z \leq \text{tol} \), return \( w_i \).
   
   (b) If \( \langle \tilde{p}_i, W_k^* H_k W_k \tilde{p}_i \rangle_z \leq 0 \), compute \( \theta > 0 \) such that \( \| W_k (w_i + \theta \tilde{p}_i) \|_x = \Delta_k - \| n_k \|_x \) and return \( w_{i+1} = w_i + \theta \tilde{p}_i \).
   
   (c) \( \tilde{a}_i = \langle \tilde{r}_i, \tilde{r}_i \rangle_z / \langle \tilde{p}_i, W_k^* H_k W_k \tilde{p}_i \rangle_z \)
   
   (d) \( w_{i+1} = w_i + \tilde{a}_i \tilde{p}_i \)
   
   (e) If \( \| W_k w_{i+1} \|_x \geq \Delta_k - \| n_k \|_x \), compute \( \theta > 0 \) such that \( \| W_k (w_i + \theta \tilde{p}_i) \|_x = \Delta_k - \| n_k \|_x \) and return \( w_{i+1} = w_i + \theta \tilde{p}_i \).
   
   (f) \( \tilde{r}_{i+1} = \tilde{r}_i + \tilde{a}_i W_k^* H_k W_k \tilde{p}_i \)
   
   (g) \( \tilde{\beta}_i = \langle \tilde{r}_{i+1}, \tilde{r}_{i+1} \rangle_z / \langle \tilde{r}_i, \tilde{r}_i \rangle_z \)
   
   (h) \( \tilde{p}_{i+1} = -\tilde{r}_{i+1} + \tilde{\beta}_i \tilde{p}_i \)

□

We now make several important observations. First, if linear systems arising in the application of \( W_k \) and \( W_k^* \) are solved iteratively, the linear operators \( W_k \) and \( W_k^* \) are being effectively applied with a certain level of inexactness. Second, in general, every iteration of the CG algorithm \([3.6.1]\) uses different inexact counterparts of \( W_k \) and \( W_k^* \).
For example, if a Krylov subspace method is used for the solution of the underlying linear systems, one cannot guarantee that the same Krylov polynomials are used in every application of $W_k$ and $W_k^*$, due to their dependence on the right-hand side of the linear system, which changes with every CG iteration. Third, if we could write down an inexact CG operator that replaces $W_k^*H_kW_k$, it would be nonsymmetric. In fact, the CG iteration would view this operator as being effectively nonlinear. Considering all these difficulties, it is nearly impossible to determine which quadratic functional (if any) is being minimized by Alg. 3.6.1 in presence of inexactness.

We should note that several studies of inexact Krylov methods and inexact preconditioned conjugate gradient methods address similar issues in the context of the solution of linear systems. Here, either the inexact operator $W_k^*H_kW_k$ is viewed as a variable Krylov operator, or $W_k$ is applied inexactly as a variable nonsingular preconditioning operator. Both change with every Krylov iteration. Nonetheless, the results of these studies are not directly applicable to the trust–region minimization context. In the first approach, the loss of symmetry in the Krylov operator does not allow us to solve the minimization problem by replacing it with a linear system. In the latter approach, the origin of $W_k$, which is a rectangular null–space operator, and not a preconditioner, is problematic. In both cases, the possibility of negative curvature and the trust–region constraint represent additional obstacles.

We propose a new inexact conjugate gradient algorithm for the solution of the tangential subproblem. This algorithm uses a full–space approach, in which the CG operator is $H_k$, and can be applied exactly. The inexactness is moved into a “preconditioning” operator, for which we can prove the existence of a fixed linear representation. This fact along with the symmetry of the CG operator $H_k$ allow us to relate the new inexact CG algorithm to the minimization of a quadratic functional.

In the inexact regime, the most significant obstacle in applying Alg. 3.6.1 to the reduced tangential subproblem is the loss of symmetry (and potentially linearity) of
the inexact reduced Hessian. This will be the case for any formulation that operates in
the reduced space $\mathcal{Z}$. We thus try to move the nonlinearities away from the Hessian
operator by considering problem (3.4) in the full space $\mathcal{X}$. We let $t = W_k w$, and
obtain the full–space reformulation of (3.4),

$$
\begin{align}
\min & \quad q_k(t) \equiv \frac{1}{2} \langle H_k t, t \rangle + \langle g_k, t \rangle + q_k^X(n_k) \\
\text{s.t.} & \quad t \in \text{Range}(W_k) \\
& \quad \|t\|_{\mathcal{X}} \leq \Delta_k - \|n_k\|_{\mathcal{X}}.
\end{align}
$$

(3.24a) (3.24b) (3.24c)

Analytically, this formulation has no advantages over the original full–space prob-
lem (3.3), however, there is a major algorithmic advantage. In particular, if we apply
a conjugate gradient scheme in which we can explicitly ensure that the constraints
(3.24b) and (3.24c) are satisfied at every iteration, we will have solved (3.24). This
can be done by treating the null–space operator $W_k$ as an inexact preconditioner, for
which we will find a fixed linear representation.

We start by introducing the variable null–space operators $W_k : \mathcal{Z} \to \mathcal{X}$ and
$W^*_k : \mathcal{X} \to \mathcal{Z}$ that represent $W_k$ and $W^*_k$, respectively, but can change at every CG
iteration. The composite operator $V_k : \mathcal{X} \to \mathcal{X}$, defined by

$$
V_k = W_k \circ W^*_k,
$$

will be used in our inexact CG algorithm. We now formulate the inexact full–space
CG algorithm with full orthogonalization, for the solution of (3.24) in presence of
inexactness.

Algorithm 3.6.2. (*Full-space CG for the solution of (3.24)*)

0. Given $\text{tol} > 0$. Let $t_0 = 0 \in \mathcal{X}$. Let $r_0 = g_k$.
1. For $i = 0, 1, 2, \ldots$
   (a) Compute $W^*_k(r_i)$. If $\|W^*_k(r_i)\|_{\mathcal{Z}} \leq \text{tol}$, return $t_i$. Stop.
   Compute $z_i = V_k(r_i) = W_k(W^*_k(r_i))$. 
(b) \( p_i = -z_i + \sum_{j=0}^{i-1} \frac{\langle z_i, H_k p_j \rangle}{\langle p_j, H_k p_j \rangle} p_j \)

(c) If \( \langle r_i, p_i \rangle_X \neq 0 \) and \( \langle p_i, H p_i \rangle_X \leq 0 \), compute \( \theta \) such that \( \text{sign}(\theta) = \text{sign}(\langle r_i, p_i \rangle_X) \) and \( \|t_i + \theta p_i\|_X = \Delta - \|n_k\|_X \), and return \( t_{i+1} = t_i + \theta p_i. \)

If \( \langle r_i, p_i \rangle_X = 0 \) and \( \langle p_i, H p_i \rangle_X < 0 \), compute \( \theta \) such that \( \|t_i + \theta p_i\|_X = \Delta - \|n_k\|_X \), and return \( t_{i+1} = t_i + \theta p_i. \)

(d) If \( \langle r_i, p_i \rangle_X = 0 \), stop and return \( t_i. \)

(e) \( \alpha_i = -\frac{\langle r_i, p_i \rangle}{\langle p_i, H_k p_i \rangle} \)

(f) \( t_{i+1} = t_i + \alpha_i p_i \)

(g) If \( \|t_{i+1}\|_X \geq \Delta - \|n_k\|_X \), compute \( \theta \) such that \( \text{sign}(\theta) = \text{sign}(\alpha_i) \) and \( \|t_i + \theta p_i\|_X = \Delta - \|n_k\|_X \), and return \( t_{i+1} = t_i + \theta p_i. \)

(h) \( r_{i+1} = H_k t_{i+1} + g_k = r_i + \alpha_i H_k p_i \)

Remark 3.6.3. If \( W_k = W_k^* \) and \( W_k^* = W_k^* \), then \( \langle r_i, p_i \rangle_X < 0. \) Since Algorithm 3.6.2 uses \( V_k = W_k \circ W_k^* \) instead of \( W_k W_k^* \), it is possible that \( \langle r_i, p_i \rangle_X \geq 0. \) This makes the check for negative curvature in Step (1c) of Algorithm 3.6.2 slightly more involved. Furthermore, if \( \langle r_i, p_i \rangle_X = 0 \), in Step (1d), then \( \langle p_i, H p_i \rangle_X \geq 0 \) and Algorithm 3.6.2 stagnates. Therefore, the stopping condition (1d) is included in Algorithm 3.6.2. Due to inexact application of \( W_k W_k^* \) we may have \( \langle r_i, p_i \rangle_X > 0. \) \( r_i = \nabla q_k(t_i) \) is not a descent direction of \( q_k \) at \( t_i \), but \( -p_i \) is. If \( \langle r_i, p_i \rangle_X > 0 \), then \( \alpha_i < 0 \), hence \( q_k \) will be reduced when moving from \( t_i \) to \( t_i + \alpha_i p_i \). If \( t_i + \alpha_i p_i \) is outside the trust-region, we can only move from \( t_i \) to \( t_i + \alpha_i p_i \) until the trust-region boundary is reached. This is accomplished in Algorithm 3.6.2. Step (1g) by choosing the sign of \( \theta \) equal to the sign of \( \alpha_i. \)

The control of inexactness in the inexact null–space operator is the subject of the next chapter. Nonetheless, several important properties of Alg. 3.6.2 can already be established. We first show the equivalence of Alg. 3.6.2 and Alg. 3.6.1 in the exact case. This property is important, as we would like to recover the convergence properties of the exact CG method if linear system solves can be performed with high accuracy. We then show that the search directions \( p_i \) are \( H_k \)-orthogonal, which further
enables us to use a standard technique to prove that every iterate of the algorithm minimizes $q$ over the set of the previously computed search directions. Finally, we show that, from the point of view of Alg. 3.6.2, the nonlinear operators $W_k$ and $W_k^*$ can be replaced with linear representations $W_{2,k}$ and $W_{1,k}^*$, respectively. The existence of $W_{2,k}$ enables us to formulate an inexact quadratic functional which is minimized by Alg. 3.6.2.

We start with the proof of equivalence of Alg. 3.6.2 and Alg. 3.6.1 in the exact case. In order to simplify the presentation, we introduce an equivalent version of Alg. 3.6.1.

Algorithm 3.6.4. (Steihaug–Toint CG for the solution of (3.4), alternate.)

0. Given $\text{tol} > 0$. Let $w_0 = 0 \in \mathcal{Z}$. Let $\tilde{r}_0 = W_k^* g_k$.

1. For $i = 0, 1, 2, ...$

   (a) $\tilde{p}_i = -\tilde{r}_i + \sum_{j=0}^{i-1} \langle \tilde{r}_i, W_k^* H_k W_k \tilde{p}_j \rangle_Z \tilde{p}_j$

   (b) If $\langle \tilde{p}_i, W_k^* H_k W_k \tilde{p}_i \rangle_Z \leq 0$, compute $\theta > 0$ such that $\|W_k(w_i + \theta \tilde{p}_i)\|_X = \Delta_k - \|n_k\|_X$ and return $w_{i+1} = w_i + \theta \tilde{p}_i$.

   (c) $\tilde{\alpha}_i = -\frac{\langle \tilde{r}_i, \tilde{p}_i \rangle_Z}{\langle \tilde{p}_i, W_k^* H_k W_k \tilde{p}_i \rangle_Z}$

   (d) $w_{i+1} = w_i + \tilde{\alpha}_i \tilde{p}_i$

   (e) If $\|W_k w_{i+1}\|_X \geq \Delta_k - \|n_k\|_X$, compute $\theta > 0$ such that $\|W_k(w_i + \theta \tilde{p}_i)\|_X = \Delta_k - \|n_k\|_X$ and return $w_{i+1} = w_i + \theta \tilde{p}_i$.

   (f) $\tilde{r}_{i+1} = W_k^* H_k W_k w_{i+1} + W_k^* g_k$

\square

We should note that the sum in Step 1a automatically truncates to its last term for $i \geq 1$, i.e.

$$
\sum_{j=0}^{i-1} \langle \tilde{r}_i, W_k^* H_k W_k \tilde{p}_j \rangle_Z \tilde{p}_j = \frac{\langle \tilde{r}_i, W_k^* H_k W_k \tilde{p}_{i-1} \rangle_Z}{\langle \tilde{p}_{i-1}, W_k^* H_k W_k \tilde{p}_{i-1} \rangle_Z} \tilde{p}_{i-1},
$$

due to an argument similar to Remark 3.6.9. All other relationships necessary to establish the equivalence of Alg. 3.6.1 and Alg. 3.6.4 can be found in standard textbooks, see e.g. [65, p.187ff] or [60, p.108ff].
Lemma 3.6.5. Let \( w_j, \tilde{p}_j, \tilde{\alpha}_j, \tilde{r}_j, j = 0, \ldots, i \), be the iterates, search directions, step sizes and gradients generated by Algorithm 3.6.1 applied to the reduced–space problem (3.4). Furthermore, let \( s_j, p_j, \alpha_j, \) and \( r_j, j = 0, \ldots, i \), be the iterates, search directions, step sizes and gradients generated by Algorithm 3.6.2 applied to the full–space problem (3.24). If \( W_k = W_k \) and \( \mathcal{W}_k = \mathcal{W}_k^* \), then

\[
\begin{align*}
  t_j &= W_k w_j, \quad p_j = W_k \tilde{p}_j, \quad \alpha_j = \tilde{\alpha}_j, \quad \text{and} \quad \tilde{r}_j = W_k^* r_j, \quad j = 0, \ldots, i.
\end{align*}
\]

Proof. We show by induction that the desired relationships hold for Alg. 3.6.2 and Alg. 3.6.4 (the latter is equivalent to Alg. 3.6.1). Let \( i = 0 \). Before entering the iteration, we have set \( 0 = t_0 = W_0 W_k w_0 \) and \( \tilde{r}_0 = W_k^* g_k = W_k^* r_0 \). Furthermore,

\[
\begin{align*}
  p_0 &= -W_k W_k^* r_0 = -W_k (W_k^* r_0) = -W_k \tilde{r}_0 = W_k \tilde{p}_0,
\end{align*}
\]

and

\[
\begin{align*}
  \alpha_0 &= -\frac{\langle r_0, p_0 \rangle_X}{\langle p_0, H_k p_0 \rangle_X} = -\frac{\langle r_0, W_k \tilde{p}_0 \rangle_X}{\langle \tilde{p}_0, H_k W_k \tilde{p}_0 \rangle_X} = -\frac{\langle W_k^* r_0, \tilde{p}_0 \rangle_Z}{\langle \tilde{p}_0, W_k^* H_k W_k \tilde{p}_0 \rangle_Z} = -\frac{\langle \tilde{r}_0, \tilde{p}_0 \rangle_Z}{\langle \tilde{p}_0, W_k^* H_k W_k \tilde{p}_0 \rangle_Z} = \tilde{\alpha}_0.
\end{align*}
\]

Assume that the claim is true for iteration \( i - 1 \). By the induction hypothesis,

\[
\begin{align*}
  t_i &= t_{i-1} + \alpha_{i-1} p_{i-1} = W_k w_{i-1} + \alpha_{i-1} W_k \tilde{p}_{i-1} = W_k (w_{i-1} + \tilde{\alpha}_{i-1} \tilde{p}_{i-1}) = W_k w_i,
\end{align*}
\]

and

\[
\begin{align*}
  \tilde{r}_i &= W_k^* H_k W_k w_i + W_k^* g_k = W_k^* (H_k (W_k w_i) + g_k) = W_k^* (H_k t_i + g_k) = W_k^* r_i.
\end{align*}
\]
Finally, as above,

\[ p_i = -W_k^* r_i + \sum_{j=0}^{i-1} \frac{\langle W_k W_k^* r_i, H_k p_j \rangle}{\langle p_j, H_k p_j \rangle} p_j = -W_k \tilde{r}_i + \sum_{j=0}^{i-1} \frac{\langle W_k \tilde{r}_i, H_k W_k \tilde{p}_j \rangle}{\langle W_k \tilde{p}_j, H_k W_k \tilde{p}_j \rangle} W_k \tilde{p}_j \]

\[ = W_k \left( -\tilde{r}_i + \sum_{j=0}^{i-1} \frac{\langle \tilde{r}_i, W_k^* H_k W_k \tilde{p}_j \rangle}{\langle \tilde{p}_j, W_k^* H_k W_k \tilde{p}_j \rangle} \tilde{p}_j \right) = W_k \tilde{p}_i, \]

and

\[ \alpha_i = \frac{\langle r_i, p_i \rangle}{\langle p_i, H_k p_i \rangle} = \frac{\langle r_i, W_k \tilde{p}_i \rangle}{\langle W_k \tilde{p}_i, H_k W_k \tilde{p}_i \rangle} = \frac{\langle W_k^* r_i, \tilde{p}_i \rangle}{\langle \tilde{p}_i, W_k^* H_k W_k \tilde{p}_i \rangle} = \frac{\langle \tilde{r}_i, \tilde{p}_i \rangle}{\langle \tilde{p}_i, W_k^* H_k W_k \tilde{p}_i \rangle} = \tilde{\alpha}_i. \]

The following lemma shows the $H_k$-conjugacy of the search directions, which will help us in proving the minimization property in Lemma 3.6.7.

**Lemma 3.6.6 (Conjugacy Property).** The search directions $p_j$, $j = 0, \ldots, i$, generated by Algorithm 3.6.2 satisfy

\[ \langle p_s, H_k p_l \rangle = 0, \]

for all $s \neq l$ with $0 \leq l, s \leq i$.

**Proof.** Let $i = 1$. Recalling that $H_k$ is self adjoint, we have

\[ \langle p_0, H_k p_1 \rangle = \langle p_1, H_k p_0 \rangle = \left\langle -\mathcal{V}_k(r_1) + \frac{\langle \mathcal{V}_k(r_1), H_k p_0 \rangle}{\langle p_0, H_k p_0 \rangle} p_0, H_k p_0 \right\rangle = -\langle \mathcal{V}_k(r_1), H_k p_0 \rangle + \langle \mathcal{V}_k(r_1), H_k p_0 \rangle = 0. \]

Assume that the claim is true for iteration $i - 1$, i.e. for $0 \leq l, s \leq i - 1$. Without loss of generality, assume $l < s$. Due to the induction hypothesis, the claim need only be
shown for the case $0 \leq l < s = i$. We obtain
\[
\langle p_l, H_k p_i \rangle_X = \langle p_i, H_k p_l \rangle_X = \left\langle -\mathcal{V}_k(r_i) + \sum_{j=0}^{i-1} \frac{\langle \mathcal{V}_k(r_i), H_k p_j \rangle_X}{\langle p_j, H_k p_j \rangle_X} p_j, H_k p_l \right\rangle_X,
\]
and notice that the last index in the sum above is $i-1$, i.e. we can apply the induction hypothesis to obtain
\[
\langle p_l, H_k p_i \rangle_X = \langle p_i, H_k p_l \rangle_X = \left\langle -\mathcal{V}_k(r_i) + \frac{\langle \mathcal{V}_k(r_i), H_k p_l \rangle_X}{\langle p_i, H_k p_l \rangle_X} p_l, H_k p_l \right\rangle_X = 0.
\]

Lemma 3.6.7 (Minimization Property). Suppose Algorithm 3.6.2 returns with $t_{i+1}$. The iterates $t_j$, $j = 1, \ldots, i$, minimize $q_k$, defined in (3.24a), over span\{\{p_0, p_1, \ldots, p_{j-1}\}\}. Moreover,
\[
q_k(t_i) < q_k(t_{i+1}) < \ldots < q_k(t_0) = 0.
\]

Proof. Let $j \in \{0, \ldots, i\}$. Using the conjugacy property, we find that
\[
q_k \left( \sum_{\ell=0}^{j} \alpha_{\ell} p_\ell \right) = \sum_{\ell=0}^{j} \alpha_{\ell}^2 \langle p_\ell, H_k p_\ell \rangle_X + \alpha_{\ell} \langle g_{\ell}, p_\ell \rangle_X.
\]
Since Algorithm 3.6.2 does not terminate in iteration $j \leq i$, we have $\langle p_\ell, H_k p_\ell \rangle_X > 0$, for $\ell = 0, 1, \ldots, j$. Thus a unique minimizer $(\alpha_{\ell}^*, \alpha_{\ell}^*, \ldots, \alpha_{j}^*)$ is given by
\[
\alpha_{\ell}^* = -\frac{\langle r_0, p_\ell \rangle_X}{\langle p_\ell, H_k p_\ell \rangle_X},
\]
and therefore $\sum_{\ell=0}^{j} \frac{-\langle r_0, p_\ell \rangle_X}{\langle p_\ell, H_k p_\ell \rangle_X} p_\ell$ minimizes $q_k$ over span\{\{p_0, p_1, \ldots, p_{j-1}\}\}. Step 1h) of Algorithm 3.6.2 implies
\[
r_\ell = r_0 - \sum_{s=0}^{\ell-1} \frac{\langle r_s, p_s \rangle_X}{\langle p_s, H_k p_s \rangle_X} H_k p_s.
\]
thus
\[
\langle r_\ell, p_\ell \rangle = \left\langle r_0 - \sum_{s=0}^{\ell-1} \frac{\langle r_s, p_s \rangle_X}{\langle p_s, H_k p_s \rangle_X} H_k p_s, p_\ell \right\rangle_X = \langle r_0, p_\ell \rangle_X,
\]
where we have used the conjugacy property. This implies that \( t_j \) minimizes \( q_k \) over \( \text{span}\{p_0, p_1, \ldots, p_{j-1}\} \) and, since, \( \alpha_j \neq 0, j = 0, \ldots, i \), (otherwise, Algorithm 3.6.2 would have returned with \( t_j, j < i + 1 \)),
\[
q_k(t_i) < \ldots < q_k(t_0) = 0. \tag{3.25}
\]

To complete the proof, we need to consider the last iterate \( t_{i+1} \) and show that \( q_k(t_{i+1}) < q_k(t_i) \).

If \( t_{i+1} \) is returned (in iteration \( i + 2 \)) in Step (1a) or in Step (1d), then using the same arguments as before, we can show that \( t_{i+1} \) minimizes \( q_k \) over \( \text{span}\{p_0, p_1, \ldots, p_i\} \) and \( q_k(t_{i+1}) < q_k(t_i) \).

If \( t_{i+1} \) is generated by Step (1c) or by Step (1g), we have
\[
q_k(t_{i+1}) = q(t_i + \theta p_i) = q(t_i) + \theta \langle g_k + H_k t_i, p_i \rangle_X + \frac{\theta^2}{2} \langle p_i, H_k p_i \rangle_X = q_k(t_i) + \theta \langle r_i, p_i \rangle_X + \frac{\theta^2}{2} \langle p_i, H_k p_i \rangle_X.
\]

If \( t_{i+1} \) is generated by Step (1c), then \( \langle p_i, H_k p_i \rangle_X \leq 0 \) and \( \theta \langle r_i, p_i \rangle_X < 0 \) or alternatively \( \langle p_i, H_k p_i \rangle_X < 0 \) and \( \theta \langle r_i, p_i \rangle_X = 0 \). In both cases we obtain \( q_k(t_{i+1}) < q_k(t_i) \). Note that since the algorithm did not terminate in the previous iteration in Step (1g), we have \( \|t_i\|_X < \Delta_k - \|n_k\|_X \) and, consequently, \( \theta \neq 0 \).

If \( t_{i+1} \) is generated by Step (1g), then \( \langle r_i, p_i \rangle_X < 0 \) and \( \langle p_i, H_k p_i \rangle_X > 0 \). The quadratic \( \varphi \mapsto \varphi \langle r_i, p_i \rangle_X + \frac{\varphi^2}{2} \langle p_i, H_k p_i \rangle_X \) strictly decreases as \( \varphi \) goes from 0 to \( \alpha_i \) computed in Step (1e). Since \( 0 < |\theta| \leq |\alpha_i| \) and \( \text{sign}(\theta) = \text{sign}(\alpha_i) \), this implies \( q_k(t_{i+1}) < q_k(t_i) \). Again, note that since the algorithm did not terminate in the previous iteration in Step (1g), we have \( \|t_i\| < \Delta_k - \|n_k\|_X \), thus \( |\theta| > 0 \).

In the remainder of this section, we turn to proving the existence of fixed linear
operators that can replace their nonlinear counterparts $W_k$ and $W_k^*$ in Alg. 3.6.2. The existence of these operators is based on Lemma 3.6.14, which states that the action of any nonlinear operator on a linearly independent set of vectors can always be interpreted as the action of some linear operator on that set of vectors. As far as Alg. 3.6.2 is concerned, there are two sets of vectors to which nonlinear operators are applied. One, associated with the operator $W_k^*$, is the set of residuals $\{r_i\}$, the other is the set $\{W_k^*(r_i)\}$, acted on by the operator $W_k$. We will prove that the set $\{r_i\}$ is linearly independent, by using Lemma 3.6.8. We will assume that the set $\{W_k^*(r_i)\}$ is linearly independent. In practice, this turns out to be a rather weak assumption. One possible explanation for this is that if $W_k = W_k^*$, the set $\{W_k^*(r_i)\}$ is not only linearly independent, but in fact orthogonal, see Lemma 3.6.11.

**Lemma 3.6.8.** The search directions $p_j$ and residuals $r_{j+1}$, $j = 0, \ldots, i$, generated by Algorithm 3.6.2 satisfy

$$\langle r_s, p_l \rangle = 0 \quad \text{for all } l, s \text{ with } 0 \leq l < s \leq i + 1.$$

**Proof.** We start the inductive argument with $i = 0$, in which case we need to consider

$$\langle r_1, p_0 \rangle_X = \left( r_0 - \frac{\langle r_0, p_0 \rangle_X}{\langle p_0, H_k p_0 \rangle_X} H_k p_0, p_0 \right)_X = \langle r_0, p_0 \rangle_X - \langle r_0, p_0 \rangle_X = 0.$$

We suppose that the claim is true for iteration $i - 1$, i.e. for $0 \leq l < s \leq i$, and show its validity for iteration $i$, i.e. the case $0 \leq l < s \leq i + 1$. The induction hypothesis reduces this case to $0 \leq l < s = i + 1$. We obtain

$$\langle r_{i+1}, p_l \rangle_X = \left( r_i - \frac{\langle r_i, p_i \rangle_X}{\langle p_i, H_k p_i \rangle_X} H_k p_i, p_i \right)_X = \langle r_i, p_i \rangle_X - \frac{\langle r_i, p_i \rangle_X}{\langle p_i, H_k p_i \rangle_X} \langle p_i, H_k p_i \rangle_X$$

$$= \begin{cases} \langle r_i, p_i \rangle_X - \langle r_i, p_i \rangle_X = 0, & \text{for } l = i, \\ 0 - \frac{\langle r_i, p_i \rangle_X}{\langle p_i, H_k p_i \rangle_X}, & \text{for } l < i, \end{cases}$$
where we have used the induction hypothesis and the conjugacy property. □

**Remark 3.6.9.** Let $V_k = V_k$ be a self-adjoint linear operator. We can now easily show the reduction of $\sum_{j=0}^{i-1} \langle V_k r_j, H_k p_j \rangle_X p_j$ to $\langle V_k r_i, H_k p_i \rangle_X p_i$. For any $j < i$, from Lemma 3.6.8 we get

$$0 = \langle r_i, p_j \rangle_X = \left( r_i, -V_k r_j + \sum_{l=0}^{j-1} \frac{\langle V_k r_j, H_k p_l \rangle_X}{\langle p_l, H_k p_l \rangle_X} p_l \right)_X,$$

which by another application of Lemma 3.6.8 implies

$$\langle r_i, V_k r_j \rangle_X = 0.$$

Since $V_k$ is self-adjoint,

$$\langle r_j, V_k r_i \rangle_X = 0$$

is also true. From Step 1h of Alg. 3.6.2 we obtain

$$\langle V_k r_i, H_k p_j \rangle_X = \frac{\langle p_j, H_k p_i \rangle_X}{\langle r_i, p_i \rangle_X} \langle V_k r_i, r_j - r_{j+1} \rangle_X = 0$$

i.e.

$$\langle V_k(r_i), H_k p_j \rangle_X = \langle V_k r_i, H_k p_j \rangle_X = 0,$$

for every iteration $i \geq 2$ and any $0 \leq j \leq i - 2$, and the claim follows.

**Lemma 3.6.10.** Let $r_j, j = 0, \ldots, i$, be residuals generated by Algorithm 3.6.2. If $r_i \neq 0$, then $r_j, j = 0, \ldots, i$, are linearly independent.

**Proof.** Suppose that the vectors $r_0, \ldots, r_i$ are linearly dependent. Then there exists a sequence of coefficients $\{\xi_j\}_{j=0}^i$ with $\sum_{j=0}^i |\xi_j| \neq 0$, i.e. not all $\xi_j$ can be zero, such that

$$\sum_{j=0}^i \xi_j r_j = 0.$$
Without loss of generality let $\xi_s$ be the first nonzero coefficient in $\{\xi_j\}_{j=0}^i$, i.e. the sum above starts at index $s$. Thus

$$0 = \left \langle \sum_{j=s}^i \xi_j r_j, p_s \right \rangle_X = \xi_s \left \langle r_s, p_s \right \rangle_X,$$

due to Lemma 3.6.8, i.e. $\left \langle r_s, p_s \right \rangle_X = 0$. This means that Alg. 3.6.2 would have terminated at iteration $s$ in Step (1c) with $\left \langle r_s, p_s \right \rangle_X = 0$, and the linear combination from above reduces to $\xi_s r_s = 0$, which is a contradiction to the assumptions $\xi_s \neq 0$ and $r_s \neq 0$.

**Lemma 3.6.11.** Let $r_j, j = 0, \ldots, i$, be the residuals generated by Algorithm 3.6.2. If $W_k = W_k^*$ and $W_k^* = W_k^*$, the vectors $W_k^* r_0, \ldots, W_k^* r_i$ are orthogonal.

**Proof.** From Lemma 3.6.8 $\left \langle r_j, p_{j_2} \right \rangle_X = 0$ for $j_2 < j_1$, thus

$$\left \langle r_{j_1}, p_{j_2} \right \rangle_X = \left \langle r_{j_1}, -V_k(r_{j_2}) + \sum_{l=0}^{j_2-1} \frac{V_k(r_{j_2})}{p_l} H_k p_l \right \rangle_X = \left \langle r_{j_1}, -V_k(r_{j_2}) \right \rangle_X = 0.$$

We have assumed $V = W_k W_k^*$, and thus obtain

$$\left \langle r_{j_1}, W_k W_k^* r_{j_2} \right \rangle_X = \left \langle W_k^* r_{j_1}, W_k^* r_{j_2} \right \rangle_Z = 0,$$

for all $j_1 \neq j_2$. \hfill $\square$

**Assumption 3.6.12.** Let $r_j, j = 0, \ldots, i$, be the residuals generated by Algorithm 3.6.2. The vectors $W_k^* (r_0), \ldots, W_k^* (r_i)$ are linearly independent.

**Remark 3.6.13.** In many cases, with help of Lemma 3.6.11 and under certain mild requirements, Assumption 3.6.12 can be eliminated. For example, in Section 4.1, we use Lemma 3.6.11 to show the invertibility of a matrix related to the quantities $W_k^* (r_j)$. Additionally, in large-scale applications of interest, it is feasible to explicitly verify Assumption 3.6.12, and use it as a termination criterion for Alg. 3.6.2.
We now state a lemma that will help us prove a main result.

**Lemma 3.6.14.** Let $\mathcal{X}$ and $\mathcal{Y}$ be Hilbert spaces, let $\mathcal{X}$ be separable, and let $V = \{v_i\}$ be a countable linearly independent set of vectors in $\mathcal{X}$. Let $\mathcal{N} : \mathcal{X} \to \mathcal{Y}$ be an arbitrary (generally nonlinear) mapping. Then there exists a fixed linear operator $\overline{\mathcal{N}} : \mathcal{X} \to \mathcal{Y}$ such that $\overline{\mathcal{N}}v_i = \mathcal{N}(v_i)$ for all $v_i \in V$.

**Proof.** Since $\mathcal{X}$ is separable and $V$ is linearly independent, there exists a countable basis $B$ for $\mathcal{X}$ such that $V \subseteq B$. Moreover, any vector $x \in \mathcal{X}$ can be uniquely written as

$$x = \sum_{b_j \in B} x_j b_j,$$

where $x_j \in \mathbb{R}$. Considering this linear expansion, the desired linear operator $\overline{\mathcal{N}}$ can be defined as follows:

$$\overline{\mathcal{N}}x = \sum_{b_j \in B} x_j \mathcal{N}(b_j),$$

for any $x \in \mathcal{X}$. First, we note that $\overline{\mathcal{N}}$ is well-defined, since $B$ is countable. Second, it has the correct property with respect to the set $V$, as for any $v_i \in V$

$$\overline{\mathcal{N}}v_i = 1 \cdot \mathcal{N}(v_i) = \mathcal{N}(v_i),$$

due to the uniqueness of the basis expansion and the fact that $v_i$ is a basis vector. Finally, we must verify that $\overline{\mathcal{N}}$ is indeed linear. We have for arbitrary real constants $\alpha$ and $\beta$

$$\overline{\mathcal{N}}(\alpha x + \beta y) = \sum_{b_j \in B} (\alpha x_j + \beta y_j) \mathcal{N}(b_j) = \alpha \sum_{b_j \in B} x_j \mathcal{N}(b_j) + \beta \sum_{b_j \in B} y_j \mathcal{N}(b_j) = \alpha \overline{\mathcal{N}}x + \beta \overline{\mathcal{N}}y.$$

**Theorem 3.6.15.** Let $r_j$, $j = 0, \ldots, i$, be residuals generated by Algorithm 3.6.2.
Then there exists a fixed linear operator $W_{1,k}$ such that

$$W_k(r_j) = W_{1,k}^* r_j, \quad j = 0, \ldots, i.$$  \hfill (3.26)

**Proof.** The proof follows from Lemma 3.6.14 and Lemma 3.6.10 \hfill \Box

**Theorem 3.6.16.** Let $r_j, j = 0, \ldots, i,$ be residuals generated by Algorithm 3.6.2 and let $W_k^*$ satisfy Assumption 3.6.12. Then there exists a fixed linear operator $W_{2,k}$ such that

$$W_k(W_k^*(r_j)) = W_{2,k} W_k^*(r_j), \quad j = 0, \ldots, i.$$  \hfill (3.27)

**Proof.** Follows directly from Lemma 3.6.14 \hfill \Box

**Remark 3.6.17.** Under the assumptions of Theorem 3.6.16 the existence of a linear operator $V_k = W_{2,k} W_{1,k}^*$ that replaces the nonlinear operator $V_k = W_k \circ W_k^*$ with respect to Alg. 3.6.2 is obvious.

Having shown the existence of a fixed linear representation of the null–space operator, we can now revisit the global convergence criteria for the tangential step computation, as given in Section 3.2.

**Lemma 3.6.18.** Under Assumption 3.6.12, every iterate $t_i$ of Alg. 3.6.2 satisfies

$$t_i \in \text{Range}(W_{2,k}).$$

**Proof.** Step (1b) of Alg. 3.6.2 yields $p_i \in \text{Range}(W_{2,k})$. Furthermore, $t_0 = 0 \in \text{Range}(W_{2,k})$. From Step (1f) we directly obtain $t_i \in \text{Range}(W_{2,k})$. \hfill \Box

Lemma 3.6.18 implies that Alg. 3.6.2 effectively sees problem (3.24) as

$$
\min \quad \frac{1}{2} \langle H_k t, t \rangle_X + \langle g_k, t \rangle_X + q_k^X(n_k) \\
\text{s.t.} \quad t \in \text{Range}(W_{2,k}) \\
\quad \|t\|_X \leq \Delta_k - \|n_k\|_X
$$  \hfill (3.28a)
Additionally, for every iterate $t_i$ of Alg. 3.6.2, there exists a $w_i \in \mathcal{Z}$, such that $t_i = W_{2,k}w_i$. Therefore, due to Lemma 3.6.7, we have that $\tilde{q}_k(w_i) < \tilde{q}_k(w_{i+1})$ with

$$
\tilde{q}_k(w) = \frac{1}{2} \langle W_{2,k}^* H_k W_{2,k} w, w \rangle_{\mathcal{Z}} + \langle W_{2,k}^* g_k, w \rangle_{\mathcal{Z}} + q^X_k (n_k),
$$

which enables us to define the inexact reduced gradient and Hessian, respectively, as follows,

$$
\tilde{W}_k^* g_k = W_{2,k}^* g_k, \quad \tilde{W}_k^* H_k W_{2,k} = W_{2,k}^* H_k W_{2,k}.
$$

The global convergence requirements for the tangential step are therefore

$$
\| W_{2,k}^* g_k - W_k^* g_k \|_{\mathcal{Z}} \leq \xi_1 \min \left\{ \| W_{2,k}^* g_k \|_{\mathcal{Z}}, \Delta_k \right\}, \quad (3.29)
$$

$$
\langle W_{2,k}^* H_k W_{2,k} w_k, w_k \rangle_{\mathcal{Z}} \leq \xi_2 \| w_k \|_{\mathcal{Z}}^2, \quad (3.30)
$$

$$
- \frac{1}{2} \langle W_{2,k}^* H_k W_{2,k} w_k, w_k \rangle_{\mathcal{Z}} - \langle W_{2,k}^* g_k, w_k \rangle_{\mathcal{Z}} \geq \kappa_4 \| W_{2,k}^* g_k \|_{\mathcal{Z}} \min \left\{ \kappa_5 \| W_{2,k}^* g_k \|_{\mathcal{Z}}, \kappa_6 \Delta_k \right\}, \quad (3.31)
$$

for constants $\xi_1, \xi_2, \kappa_4, \kappa_5, \kappa_6 > 0$, independent of $k$. These conditions can be satisfied by carefully controlling the size of $\| W_{2,k} - W_k \|$ and $\| W_{2,k} - W_{1,k} \|$. The next chapter presents the analysis for the case in which $W_k$ is an orthogonal null–space projector.
Chapter 4

Inexact Linear System Solves in an SQP Algorithm with Null–Space Projections

This chapter focuses on the derivation of a concrete SQP algorithm with inexactness control for the case in which the null–space representation $W_k$ is an orthogonal projector. The projections in the tangential step computation are applied via an iterative solution of augmented systems discussed in Section 2.1.1. The algorithm efficiently manages the stopping criteria for iterative linear system solves. The developed stopping criteria are dynamically adjusted based on the current progress toward a KKT point, trade gains in feasibility for gains in optimality and vice versa, can be easily implemented, and are sufficient to guarantee first–order global convergence of the SQP algorithm. As a side benefit, they allow for a rigorous integration of preconditioners for KKT systems.

Section 4.1 specializes the tangential step convergence requirements (3.6)–(3.8) to the projector case. A major contribution in our work is in the derivation of practical mechanisms that are sufficient for (3.6)–(3.8). Section 4.2 specializes the quasi–normal step convergence requirements (3.1)–(3.2) to the projector case. Section 4.3 addresses the boundedness of the sequence of Lagrange multipliers. Section 4.4 interprets the convergence requirements (3.11)–(3.14). The trust–region SQP algorithm with inexactness control in augmented system solves is stated in Section 4.5.
4.1 Tangential Step

This section discusses the special case in which $Z = X$ and the exact null-space operator $W_k$ is an orthogonal projector, satisfying

$$W_k^* = W_k = W_k W_k.$$  \hspace{1cm} (4.1)

This case is motivated by the full–space CG approach in [19, 47], which allows the use of KKT preconditioners.

As in the previous section, we formulate an inexact algorithm that in exact arithmetic reduces to Alg. 3.6.1. In this particular case, due to property (4.1), the resulting Alg. 4.1.1 is a trivial modification of Alg. 3.6.2.

Algorithm 4.1.1. (Projected inexact CG with full orthogonalization.)

0. Given $\text{tol} > 0$. Let $t_0 = 0 \in X$. Let $r_0 = g_k$.

1. For $i = 0, 1, 2, ...$
   (a) Compute $z_i = W_k(r_i)$. 
      If $\|z_i\|_X \leq \text{tol}$, return $t_i$. Stop.
   (b) $p_i = -z_i + \sum_{j=0}^{i-1} \frac{\langle z_i, H_k p_j \rangle}_X}{\langle p_j, H_k p_j \rangle}_X p_j$
   (c) If $\langle r_i, p_i \rangle_X \neq 0$ and $\langle p_i, H p_i \rangle_X \leq 0$, compute $\theta$ such that sign($\theta$) = sign($\langle r_i, p_i \rangle_X$) and $\|t_i + \theta p_i\|_X = \Delta_k - \|n_k\|_X$, and return $t_{i+1} = t_i + \theta p_i$.
      If $\langle r_i, p_i \rangle_X = 0$ and $\langle p_i, H p_i \rangle_X < 0$, compute $\theta$ such that $\|t_i + \theta p_i\|_X = \Delta_k - \|n_k\|_X$, and return $t_{i+1} = t_i + \theta p_i$.
   (d) If $\langle r_i, p_i \rangle_X = 0$, stop and return $t_i$.
   (e) $\alpha_i = -\frac{\langle r_i, p_i \rangle_X}{\langle p_i, H_k p_i \rangle_X}$
   (f) $t_{i+1} = t_i + \alpha_i p_i$
   (g) If $\|t_{i+1}\|_X \geq \Delta_k - \|n_k\|_X$, compute $\theta$ such that sign($\theta$) = sign($\alpha_i$) and $\|t_i + \theta p_i\|_X = \Delta_k - \|n_k\|_X$, and return $t_{i+1} = t_i + \theta p_i$.
   (h) $r_{i+1} = H_k t_{i+1} + g_k = r_i + \alpha_i H_k p_i$

□
If $W_k$ is an orthogonal null–space projector whose application involves iterative linear system solves, then following the augmented system approach, every projection $z_i$ in Step (1a) of Alg. 4.1.1 satisfies

$$
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z_i \\
y
\end{pmatrix} =
\begin{pmatrix}
0 \\
r_i 
\end{pmatrix} + \begin{pmatrix}
e_i^1 \\
e_i^2
\end{pmatrix}.
$$

(4.2)

Simply put, the main goal of this section is to develop a practical mechanism that controls global convergence requirements for the tangential step computation by controlling the size of $\|e_i\|_{X \times Y}$.

First, we state a few necessary remarks. We can immediately conclude that Alg. 4.1.1 will satisfy Lemma 3.6.6 (conjugacy property), Lemma 3.6.7 (minimization property), and Lemma 3.6.10 (linear independence of residual vectors). The latter gives us the following existence theorem.

**Theorem 4.1.2.** Let $r_j$, $j = 0, \ldots, i$, be residuals generated by Algorithm 4.1.1. Then there exists a fixed linear operator $\tilde{W}_k$ such that

$$W_k(r_j) = \tilde{W}_k r_j, \quad j = 0, \ldots, i. \tag{4.3}$$

Analogous to Lemma 3.6.18 every iterate $t_i$ of Alg. 4.1.1 satisfies

$$t_i \in \text{Range}(\tilde{W}_k),$$

thus Alg. 4.1.1 effectively sees problem (3.24) as

$$\min \frac{1}{2} \langle H_k t, t \rangle_X + \langle g_k, t \rangle_X + q_k^X(n_k) \tag{4.4a}$$

s.t. $t \in \text{Range}(\tilde{W}_k)$ \hspace{1cm} \tag{4.4b}

$$\|t\|_X \leq \Delta_k - \|n_k\|_X. \tag{4.4c}$$
Therefore, we can replace the exact reduced–space quadratic model with

$$\tilde{q}_k(w) = \frac{1}{2} \left( \tilde{W}_k^* H_k \tilde{W}_k w, w \right)_X + \left( \tilde{W}_k^* g_k, w \right)_X + q_k^X(n_k).$$

(4.5)

Consequently, the inexact reduced gradient and Hessian are given by

$$\tilde{W}_k^* g_k = \tilde{W}_k^* g_k, \quad \text{and} \quad \tilde{W}_k^* H_k \tilde{W}_k = \tilde{W}_k^* H_k \tilde{W}_k.$$

It should be noted that $\tilde{W}_k$ is, in general, neither self–adjoint nor a projector. Finally, for completeness, we restate the SQP convergence requirements,

$$\|\tilde{W}_k^* g_k - W_k g_k\|_X \leq \xi_1 \min \left\{ \|\tilde{W}_k^* g_k\|_X, \Delta_k \right\},$$

(4.6)

for some $\xi_1 > 0$ independent of $k$,

$$\left( \tilde{W}_k^* H_k \tilde{W}_k w_k, w_k \right)_X \leq \xi_2 \|w_k\|^2_X,$$

(4.7)

for some $\xi_2 > 0$ independent of $k$, and

$$- \frac{1}{2} \left( \tilde{W}_k^* H_k \tilde{W}_k w_k, w_k \right)_X - \left( \tilde{W}_k^* g_k, w_k \right)_X \geq \kappa_4 \|\tilde{W}_k^* g_k\|_X \min \left\{ \kappa_5 \|\tilde{W}_k^* g_k\|_X, \kappa_6 \Delta_k \right\},$$

(4.8)

for constants $\kappa_4, \kappa_5, \kappa_6 > 0$, independent of $k$.

### 4.1.1 Satisfaction of General Convergence Requirements for Inexactly Computed Tangential Steps

In order to analyze the inexactness in the application of the null–space operator, we seek an explicit representation of $\tilde{W}_k$. Suppose that $W_k$ is applied $i + 1$ times in Alg. 4.1.1. As vectors $r_j$, $j = 0, 1, ..., i$, are linearly independent, the matrix $R_i : \mathbb{R}^{i+1} \rightarrow X$, \[\begin{align*}
\end{align*}[/]
given by

\[ R_i = [r_0, r_1, \ldots, r_i] \]  

(4.9)

has full column rank. Furthermore, we introduce matrices \( Y_i : \mathbb{R}^{i+1} \to X \) and \( \tilde{Y}_i : \mathbb{R}^{i+1} \to X \), given by

\[ Y_i = [W_k r_0, W_k r_1, \ldots, W_k r_i], \quad \tilde{Y}_i = [W_k (r_0), W_k (r_1), \ldots, W_k (r_i)]. \]  

(4.10)

We have shown that there exists a linear operator \( \tilde{W}_k \) such that

\[ \tilde{W}_k R_i = \tilde{Y}_i. \]

Any linear operator that satisfies this condition could be chosen as the explicit representation of \( \tilde{W}_k \). For example, one possible choice, involving the pseudo inverse of \( R_i \), is as follows,

\[ \tilde{W}_k = W_k + (\tilde{Y}_i - Y_i)(R_i^* R_i)^{-1} R_i^*. \]

In [16], Brown uses this idea to analyze the local convergence of inexact–Newton/finite-difference methods. In our case, the expression

\[ \tilde{W}_k = W_k + (\tilde{Y}_i - Y_i)(\tilde{Y}_i^* R_i)^{-1} \tilde{Y}_i^* \]  

(4.11)

seems a better choice, provided that the inverse of \( \tilde{Y}_i^* R_i \) exists. After establishing sufficient conditions for the invertibility of \( \tilde{Y}_i^* R_i \), the explicit representation (4.11) will be used for the remainder of this section.

We briefly address the use (4.11) as the chosen explicit representation of \( \tilde{W}_k \). To simplify the notation, we introduce the diagonal matrix \( D_i \in \mathbb{R}^{i+1 \times i+1} \), with

\[ D_i = \text{diag}(\|W_k (r_0)\|_X, \ldots, \|W_k (r_i)\|_X). \]

The advantage of representation (4.11) is that if exact linear system solves are used
in Alg. 4.1.1 then the matrix $\tilde{Y}_i^* R_i$ simplifies to

$$\tilde{Y}_i^* R_i = D_i^2,$$

i.e. it is an invertible diagonal matrix. This is due to $W_k^* = W_k = W_k W_k$ and Lemma 3.6.11, which shows the orthogonality of projected residual vectors $W_k r_0, ..., W_k r_i$. With inexact linear system solves, we hope to control, in theory and in practice, the loss of orthogonality between the projected residual vectors by monitoring how much $\tilde{Y}_i^* R_i$ deviates from $D_i^2$.

The following three lemmas are necessary to prove the main results. In what follows, we denote by $\| \cdot \|_{L(R^n, X)}$ the operator norm induced by the natural inner product of the Hilbert space $X$ and the dot (scalar) product in the Euclidian space $\mathbb{R}^n$, e.g. the norm of the previously defined matrix $Y_i$ is given by

$$\|Y_i\|_{L(R^{i+1}, X)} = \sup_{x \in \mathbb{R}^{i+1}, x \neq 0} \frac{\|Y_i x\|_X}{\|x\|_2},$$

with $\|x\|_2 = \sqrt{x^T x}$. We analogously define $\| \cdot \|_{L(X, \mathbb{R}^n)}$. To simplify the notation, we denote the Euclidian matrix norm $\| \cdot \|_{L(\mathbb{R}^n, \mathbb{R}^n)}$ by $\| \cdot \|_2$.

**Lemma 4.1.3.** Suppose that Algorithm 4.1.1 terminates after $i + 1$ applications of the inexact operator $W_k$ on the residual vectors given by the first $i$ columns of $R_i$. There exists a matrix $\hat{S}_i \in \mathbb{R}^{i+1 \times i+1}$ such that

$$\tilde{Y}_i^T R_i = D_i(I_i + \hat{S}_i)D_i.$$

If $\|\hat{S}_i\|_2 \leq \sigma < 1$, then $\tilde{Y}_i^* R_i$ is invertible and there exists a matrix $S_i \in \mathbb{R}^{i+1 \times i+1}$ such that

$$(\tilde{Y}_i^T R_i)^{-1} = D_i^{-1}(I_i + S_i)D_i^{-1},$$

with $\|S_i\|_2 \leq \sigma/(1 - \sigma)$. 

Proof. We can write
\[ \tilde{Y}_i^T R_i = D_i^2 + E_i. \]

If we define \( \hat{S}_i = D_i^{-1}E_iD_i^{-1} \) we obtain \( \tilde{Y}_i^T R_i = D_i(I_i + \hat{S}_i)D_i \).

If \( \|\hat{S}_i\|_2 < 1 \), then \( (I_i + \hat{S}_i) \) is invertible and \( (I_i + \hat{S}_i)^{-1} = I_i + S_i \) with \( S_i = -\hat{S}_i(I_i + \hat{S}_i)^{-1} \) and \( \|S_i\|_2 \leq \|\hat{S}_i\|_2/(1 - \|\hat{S}_i\|_2) \). This implies the second part. \( \square \)

Lemma 4.1.4. Let \( U_i \) be a matrix, such that
\[ U_i^* = \tilde{Y}_iD_i^{-1} = \left[ \frac{W_k(r_0)}{||W_k(r_0)||_X}, \ldots, \frac{W_k(r_i)}{||W_k(r_i)||_X} \right]. \]

If \( \tilde{Y}_i^* R_i \) is invertible and \( (\tilde{Y}_i^* R_i)^{-1} = D_i^{-1}(I_i + S_i)D_i^{-1} \), then
\[ \|\tilde{W}_k - W_k\|_{L(X)} \leq (1 + \|S_i\|_2)\|\tilde{Y}_i - Y_i\|_{L(\mathbb{R}^{i+1},X)}\|U_i\|_{L(X,\mathbb{R}^{i+1})}. \]

Proof. By definition (4.11) of \( \tilde{W}_k \), we have
\[ \tilde{W}_k - W_k = (\tilde{Y}_i - Y_i)(\tilde{Y}_i^* R_i)^{-1}\tilde{Y}_i^*. \]

Hence
\[ \|\tilde{W}_k - W_k\|_{L(X)} = \|\tilde{Y}_i - Y_i\|_{L(X)} \tilde{D}_i^{-1} (I_i + S_i) \tilde{D}_i^{-1} \tilde{Y}_i^* \|_{L(X)} \]
\[ = \|\tilde{Y}_i - Y_i\|_{L(X)} \|D_i^{-1} (I_i + S_i) U_i\|_{L(X)} \]
\[ \leq \|\tilde{Y}_i - Y_i\|_{L(X)} \|D_i^{-1} \|_{L(\mathbb{R}^{i+1},X)} \|U_i\|_{L(X,\mathbb{R}^{i+1})} \]
\[ + \|\tilde{Y}_i - Y_i\|_{L(X)} \|D_i^{-1} \|_{L(\mathbb{R}^{i+1},X)} \|S_i\|_2 \|U_i\|_{L(X,\mathbb{R}^{i+1})}, \]

which implies the assertion. \( \square \)

Lemma 4.1.5. Let \( A : \mathbb{R}^{i+1} \to X \) be a matrix with \( A = [a_0, \ldots, a_i] \). Then
\[ \|A\|_{L(\mathbb{R}^{i+1},X)} \leq \sqrt{i + 1} \max_{j=0,\ldots,i} \|a_j\|_X. \]
Proof. The proof is a simple consequence of the definition of induced matrix norms. We have

$$\|A\|_{L(\mathbb{R}^{i+1},\mathcal{X})} = \sup_{x \neq 0} \frac{\| \sum_{j=0}^{i} x_j a_j \|_{\mathcal{X}}}{\|x\|_2} \leq \sup_{x \neq 0} \frac{\sum_{j=0}^{i} |x_j| \|a_j\|_{\mathcal{X}}}{\|x\|_2} \leq \max_{j=0,\ldots,i} |x_j| \|a_j\|_{\mathcal{X}} \leq \sqrt{i+1} \max_{j=0,\ldots,i} |a_j| \|x\|.$$

We can now state the theorems that introduce conditions on inexactness in the tangential step computation, such that the global convergence of the SQP algorithm is guaranteed. Under the assumptions of Lemma 4.1.3, we have confirmed the existence of the explicit operator $\tilde{W}_k$ given by (4.11). Therefore, we can use the notation $\tilde{W}_k r_j$ in place of $W_k(r_j)$.

Theorem 4.1.6. Suppose that Alg. 4.1.1 terminates after $i + 1$ applications of the inexact operator $W_k$. If there exist $C_1, C_2 > 0$ such that

$$\|\tilde{W}_k r_j - W_k r_j\|_{\mathcal{X}} \leq C_1 \min \left\{ \frac{\|\tilde{W}_k g_k\|_{\mathcal{X}}}{\|g_k\|_{\mathcal{X}}}, \frac{\Delta_k}{\|g_k\|_{\mathcal{X}}} \right\} \|\tilde{W}_k r_j\|_{\mathcal{X}}, \quad j = 0, \ldots, i, \quad (4.12)$$

and if $Y^*_i R_i$ is invertible and $(Y^*_i R_i)^{-1} = D^{-1}_i (I_i + S_i) D^{-1}_i$, then

$$\|\tilde{W}_k^* g_k - W_k g_k\|_{\mathcal{X}} \leq (1 + \|S_i\|_2)(i + 1)C_1 \max \{C_1, C_2\} \min \left\{ \|\tilde{W}_k^* g_k\|_{\mathcal{X}}, \Delta_k \right\}. \quad (4.14)$$

Proof. We have

$$\|\tilde{W}_k^* g_k - W_k g_k\|_{\mathcal{X}} \leq \|\tilde{W}_k^* - W_k\|_{L(\mathcal{X})} \|g_k\|_{\mathcal{X}} = \|\tilde{W}_k - W_k\|_{L(\mathcal{X})} \|g_k\|_{\mathcal{X}} = \|\tilde{W}_k - W_k\|_{L(\mathcal{X})} \|g_k\|_{\mathcal{X}}.$$
Thus, from Lemma 4.1.4, we obtain

$$\|\tilde{W}_k^* g_k - W_k g_k\|_X \leq (1 + \|S_i\|_2) \|U_i\|_{L(\mathbb{R}^{i+1}, X)} \|(Y_i - Y_i) D_i^{-1}\|_{L(\mathbb{R}^{i+1}, X)} \|g_k\|_X.$$ 

Since $$\left\| \left( \frac{\tilde{W}_k r_i}{\|\tilde{W}_k r_i\|_X} \right) \right\|_X = 1$$, Lemma 4.1.5 and $$\|U_i\|_{L(X, \mathbb{R}^{i+1})} = \|U_i^*\|_{L(\mathbb{R}^{i+1}, X)}$$ imply $$\|U_i\|_{L(X, \mathbb{R}^{i+1})} \leq \sqrt{i + 1}$$.

Assumptions (4.12) and (4.13) yield

$$\frac{\|\tilde{W}_k r_i - W_k r_i\|_X}{\|\tilde{W}_k r_i\|_X} \leq C_1 \max\{1, C_2\} \min\left\{ \frac{\|\tilde{W}_k^* g_k\|_X}{\|g_k\|_X}, \frac{\Delta_k}{\|g_k\|_X} \right\}.$$ 

Since

$$(Y_i - Y_i) D_i^{-1} = \left[ \frac{\tilde{W}_k r_0 - W_k r_0}{\|\tilde{W}_k r_0\|_X}, \ldots, \frac{\tilde{W}_k r_i - W_k r_i}{\|\tilde{W}_k r_i\|_X} \right],$$

Lemma 4.1.5 implies

$$\|(Y_i - Y_i) D_i^{-1}\|_{L(\mathbb{R}^{i+1}, X)} \leq \sqrt{i + 1} C_1 \max\{1, C_2\} \min\left\{ \frac{\|\tilde{W}_k^* g_k\|_X}{\|g_k\|_X}, \frac{\Delta_k}{\|g_k\|_X} \right\}.$$ 

Combining the previous estimates gives the desired result.

We see that (4.14) implies (4.6) if the number of iterations $i$ in Alg. 4.1.1 is bounded for all SQP iterations $k$ and if $\|S_i\|_2$ is uniformly bounded for all SQP iterations. Since the residuals in Alg. 4.1.1 are linearly independent (see Lemma 3.6.10), Alg. 4.1.1 in practice terminates after a finite number of iterations. In addition, we commonly stop Alg. 4.1.1 after at most $i_{\text{max}}$ iterations, where $i_{\text{max}}$ is much smaller than the dimension of the discrete representation of the Hilbert space $\mathcal{X}$. The norms of $S_i$ can be easily monitored and Alg. 4.1.1 can be truncated if the norm of $S_i = D_i(Y_i^* R_i)^{-1}D_i - I_i$ exceeds a threshold. If the number of iterations $i$ is small, then for large scale problems the computation of $S_i$ and $\|S_i\|_2$ is very inexpensive relative to the application of $W_k$ and execution of other steps in Alg. 4.1.1. Hence, Theorem 4.1.6 provides a way to enforce (4.6). Of course, we still have to address how (4.12) and (4.13) can be
satisfied. This will be done in the next subsection.

**Theorem 4.1.7.** Suppose that Alg. 4.1.1 terminates after \(i + 1\) applications of the inexact operator \(W_k\). If

\[
\frac{\|\widetilde{W}_kr_j - W_kr_j\|_X}{\|\widetilde{W}_kr_j\|_X} \leq C_3, \quad j = 0, \ldots, i,
\]

(4.16)

and if \(\widetilde{Y}_i^* R_i\) is invertible and \((\widetilde{Y}_i^* R_i)^{-1} = D_i^{-1}(I_i + S_i)D_i^{-1}\), then

\[
\|\widetilde{W}_k\|_{L(X)} \leq C_3(1 + \|S_i\|_2)(i + 1) + \|W_k\|_{L(X)}.
\]

(4.17)

**Proof.** From Lemma 4.1.4 and \(\|U_i\|_{L(X; R^{i+1})} \leq \sqrt{i + 1}\) (see proof of Theorem 4.1.6), we have

\[
\|\tilde{W}_k - W_k\|_{L(X)} \leq (1 + \|S_i\|_2)\sqrt{i + 1}\|\widetilde{Y}_i - Y_i\|_{D_i^{-1}l_{L(R^{i+1},X)}}.
\]

Lemma 4.1.5 (4.15), and (4.16) imply

\[
\|\widetilde{Y}_i - Y_i\|_{D_i^{-1}l_{L(R^{i+1},X)}} \leq C_3\sqrt{i + 1}.
\]

Hence

\[
\|\widetilde{W}_k - W_k\|_{L(X)} \leq C_3(1 + \|S_i\|_2)(i + 1).
\]

(4.18)

Finally, with \(\|\widetilde{W}_k - W_k\|_{L(X)} \geq \|\widetilde{W}_k\|_{L(X)} - \|W_k\|_{L(X)}\), we get the desired estimate. \(\Box\)

We see that (4.17) implies (4.7) if \(\{\|W_k\|_{L(X)}\}\) and \(\{\|H_k\|_{L(X)}\}\) are bounded, which is ensured by the general assumptions (A5) and (A6), if the number of iterations \(i \) in Alg. 4.1.1 is bounded for all SQP iterations \(k\), and if \(\|S_i\|_2\) is uniformly bounded for all SQP iterations. The latter two conditions were already discussed after the proof of Theorem 4.1.6. Hence, Theorem 4.1.7 provides a way to enforce (4.7). Of course, we still have to address how (4.16) can be satisfied. This will be done in the next
subsection.

The next result concerns the condition (4.8). Recall that we compute the tangential step $t_k$ by applying Alg. 4.1.1 to (4.4). If Algorithm 4.1.1 stops with $t_{i+1}$, then the tangential step is $t_k = t_{i+1}$. Here recall that the index $k$ refers to the outer SQP iteration, whereas the index $i$ refers to the iteration counter in Alg. 4.1.1. As we have seen at the beginning of this section, the iterates generated by Alg. 4.1.1 satisfy $t_i \in \text{Range}(\widetilde{W}_k)$, hence there exists $w_k$ with $t_k = \widetilde{W}_k w_k$.

Theorem 4.1.8. Suppose that Alg. 4.1.1 terminates with $t_{i+1}$. Set $t_k = t_{i+1}$ and define $w_k$ implicitly through $t_k = \widetilde{W}_k w_k$. If the first iterate $t_1$ generated by Alg. 4.1.1 satisfies

$$-\frac{1}{2} \langle H_k t_1, t_1 \rangle_X - \langle g_k, t_1 \rangle_X \geq \bar{\kappa}_4 \|\widetilde{W}_k g_k\|_X \min \left\{ \bar{\kappa}_5 \|\widetilde{W}_k g_k\|_X, \bar{\kappa}_6 \Delta_k \right\},$$

(4.19)

and if

$$\|\widetilde{W}_k g_k\|_X \geq C_4 \|\widetilde{W}_k^* g_k\|_X,$$

(4.20)

for $\bar{\kappa}_4, \bar{\kappa}_5, \bar{\kappa}_6 > 0$ and $C_4 > 0$ independent of $k$, then there exist $\kappa_4, \kappa_5, \kappa_6 > 0$, independent of $k$, such that (4.8) holds.

Proof. Assumptions (4.19) and (4.20) immediately yield

$$-\frac{1}{2} \langle H_k t_1, t_1 \rangle_X - \langle g_k, t_1 \rangle_X \geq \bar{\kappa}_4 C_4 \|\widetilde{W}_k^* g_k\|_X \min \left\{ \bar{\kappa}_5 \|\widetilde{W}_k^* g_k\|_X, \bar{\kappa}_6 \Delta_k \right\}.$$

The result now follows from Lemma 3.6.7 and $t_k = \widetilde{W}_k w_k$. \qed

If $\widetilde{W}_k = W_k$, then the first iterate $t_1$ in Alg. 4.1.1 is the solution of

$$\min_a \quad \frac{1}{2} \langle H_k t, t \rangle_X - \langle g_k, t \rangle_X$$

s.t. $t = -a W_k g_k$

$$\|t\|_X \leq \Delta_k - \|n_k\|_X.$$
It is well known that (see, e.g., [22, Thm. 6.3.1], [60, Lemma 4.5] and recall that \( W_k = W_k^* = W_k^2 \))

\[
-\frac{1}{2} \langle H_k t_1, t_1 \rangle - \langle g_k, t_1 \rangle \geq \frac{1}{2} \|W_k g_k\|_\mathcal{X} \min \left\{ \frac{\|W_k g_k\|_\mathcal{X}}{\mu_k}, \Delta_k - \|n_k\|_\mathcal{X} \right\}
\]

(4.21)

where \( \mu_k = 1 + \langle H_k W_k g_k, W_k g_k \rangle_\mathcal{X}/\|W_k g_k\|_\mathcal{X}^2 \). Since \( \|n_k\|_\mathcal{X} \leq \zeta \Delta_k \) (cf., (2.17b)), \( \Delta_k - \|n_k\|_\mathcal{X} \geq (1 - \zeta) \Delta_k \). Hence, (4.19) is satisfied if \( \tilde{W}_k = W_k \) and if \( \{\|H_k\|_{L(\mathcal{X})}\} \) is bounded. It is easy and computationally inexpensive to verify the condition (4.19) with \( \tilde{\kappa}_4 \leq 1/2, \tilde{\kappa}_5^{-1} = 1 + \langle H_k \tilde{W}_k g_k, \tilde{W}_k g_k \rangle_\mathcal{X}/\|\tilde{W}_k g_k\|_\mathcal{X}^2 \), and \( \tilde{\kappa}_6 = 1 - \zeta \), as the required quantities are readily available. Note that \( \tilde{W}_k g_k = z_0 = -p_0 \) in Alg. 4.1.1. Hence (4.19) with the previous choice of parameters reads

\[
-\frac{1}{2} \langle H_k t_1, t_1 \rangle - \langle g_k, t_1 \rangle \geq \frac{1}{2} \|z_0\|_\mathcal{X} \min \left\{ \|z_0\|_\mathcal{X}/(1 + \langle H_k z_0, z_0 \rangle_\mathcal{X}/\|z_0\|_\mathcal{X}^2), (1 - \zeta) \Delta_k \right\}.
\]

(4.22)

The next lemma shows that (4.20) follows from the assumptions in Theorem 4.1.6 (recall that \( r_0 = g_k \) in Alg. 4.1.1). More precisely, we have the following result.

**Lemma 4.1.9.** If there exists \( C_1 > 0 \) such that

\[
\|\tilde{W}_k g_k - W_k g_k\|_\mathcal{X} \leq C_1 \min \left\{ \frac{\|\tilde{W}_k g_k\|_\mathcal{X}}{\|g_k\|_\mathcal{X}}, \frac{\Delta_k}{\|g_k\|_\mathcal{X}^2} \right\} \|\tilde{W}_k g_k\|_\mathcal{X}
\]

(4.23)

and if \( \bar{Y}_i^T R_i \) is invertible and \( (\bar{Y}_i^T R_i)^{-1} = D_i^{-1}(I_i + S_i)D_i^{-1} \), then

\[
\|\tilde{W}_k g_k\|_\mathcal{X} \geq \frac{1}{1 + 2(1 + \|S_i\|_2)(i + 1)C_1} \|\tilde{W}_k^* g_k\|_\mathcal{X}.
\]
Proof. We have (recall that $W_k = W_k^T$)

$$
\|\widetilde{W}_k^* g_k\|_X \leq \|\widetilde{W}_k^* - \widetilde{W}_k\|_{L(X)} \|g_k\|_X + \|\widetilde{W}_k g_k\|_X
$$

$$
\leq \left(\|\widetilde{W}_k^* - W_k\|_{L(X)} + \|W_k - W_k\|_{L(X)}\right) \|g_k\|_X + \|\widetilde{W}_k g_k\|_X
$$

$$
= 2\|\widetilde{W}_k - W_k\|_{L(X)} \|g_k\|_X + \|\widetilde{W}_k g_k\|_X.
$$

Using the same arguments as in the proof of Theorem 4.1.6 we find that

$$
\|\widetilde{W}_k^* g_k\|_X \leq 2(1 + \|S_i\|_2) (i + 1) C_1 \min\left\{\|\widetilde{W}_k g_k\|_X, \Delta_k\right\} + \|\widetilde{W}_k g_k\|_X.
$$

This implies the desired result. \qed

4.1.2 Practical Convergence Requirements for Inexactly Computed Tangential Steps

In the previous sections we established theoretical convergence requirements that must be satisfied in the computation of the tangential step with inexact linear system solves. In this section we reexamine these requirements from a practical standpoint.

Condition (4.12)

Here we devise a practical approach for ensuring that

$$
\|\widetilde{W}_k r_i - W_k r_i\|_X \leq C_1 \min\left\{\|\widetilde{W}_k g_k\|_X, \Delta_k\right\} \|\widetilde{W}_k r_i\|_X,
$$

at every iteration $i$ of Alg. 4.1.1. We focus on the case when $\widetilde{W}_k$ is applied through augmented system solves, as in (4.2). Let $G_k : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ be the composite linear operator

$$
G_k = \begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
$$
and let $P : \mathcal{X} \rightarrow \mathcal{X} \times \mathcal{Y}$ be a fixed identity prolongation operator such that

$$
P_{ri} = \begin{pmatrix} r_i \\
0 \end{pmatrix}, \quad P^T \begin{pmatrix} z_i \\ y \end{pmatrix} = z_i.
$$

Then, with exact linear system solves, every projection $z_i$ in Step (1a) of Alg. 4.1.1 is computed as follows,

$$
z_i = W_k r_i = P^T G_k^{-1} P r_i.
$$

A generic iterative linear system solver does not solve the system $G_k v_i = P r_i$ exactly. Instead, assuming that it uses an invertible left preconditioner $M_k : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{X} \times \mathcal{Y}$ to accelerate its convergence, it returns a solution vector $v_i$ satisfying

$$
M_k^{-1} G_k v_i = M_k^{-1} P r_i + e_i,
$$

where $e_i \in \mathcal{X} \times \mathcal{Y}$ is known as the preconditioned residual vector. Solving for $v_i$ above gives

$$
z_i = \tilde{W}_k r_i = P^T G_k^{-1} (P r_i + M_k e_i),
$$

which implies

$$
\tilde{W}_k r_i - W_k r_i = P^T G_k^{-1} M_k e_i. \tag{4.24}
$$

Therefore, the iterative augmented system solve can be stopped at iteration $\ell_i$ if

$$
\|e_i(\ell_i)\|_{\mathcal{X} \times \mathcal{Y}} \leq \min \left\{ \frac{\|\tilde{W}_k g_k\|_{\mathcal{X}}}{\|g_k\|_{\mathcal{X}}}, \frac{\Delta_k}{\|g_k\|_{\mathcal{X}}} \right\} \|z_i(\ell_i)\|_{\mathcal{X}}, \tag{4.25}
$$

as in this case,

$$
\|\tilde{W}_k r_i - W_k r_i\|_{\mathcal{X}} \leq \|G_k^{-1} M_k\|_{L(\mathcal{X} \times \mathcal{Y})} \min \left\{ \frac{\|\tilde{W}_k g_k\|_{\mathcal{X}}}{\|g_k\|_{\mathcal{X}}}, \frac{\Delta_k}{\|g_k\|_{\mathcal{X}}} \right\} \|\tilde{W}_k r_i\|_{\mathcal{X}},
$$
and we can assume that $G_k^{-1}$ and $M_k$ are bounded (boundedness of $G_k^{-1}$ follows from the boundedness properties of $c_x(x)$ and $(c_x(x)c_x(x)^*)^{-1}$; boundedness of $M_k$ is an explicit, although very reasonable assumption).

In the first iteration of Alg. [4.1.1], we have $z_0 = \tilde{W}_k g_k$, thus we monitor the preconditioned residual at every iteration of the iterative augmented system solve, and stop the solve at iteration $\ell_0$ if

$$\|e_0^{(\ell_0)}\|_{X \times Y} \leq \min \left\{ \frac{\|z_0^{(\ell_0)}\|_X}{\|g_k\|_X}, \frac{\Delta_k}{\|g_k\|_X} \right\} \|z_0^{(\ell_0)}\|_X. \quad (4.26)$$

We should note that from the point of view of the iterative linear system solver, this is a fairly intrusive stopping condition. Nonetheless, many popular iterative solver libraries, in particular the Trilinos [66] parallel Krylov solver package AztecOO [56], as well as PETSc’s [2] parallel solvers allow for an efficient implementation of such stopping criteria. An alternative to (4.26) would be to perform the augmented solve with a fixed high accuracy in the first iteration of Alg. [4.1.1].

For subsequent iterations of Alg. [4.1.1], we use the value $\|z_0^{(\ell_0)}\|$, and terminate the augmented system solve at iteration $\ell_i$ if

$$\|e_i^{(\ell_i)}\|_{X \times Y} \leq \min \left\{ \frac{\|z_0^{(\ell_0)}\|_X}{\|g_k\|_X}, \frac{\Delta_k}{\|g_k\|_X} \right\} \|z_i^{(\ell_i)}\|_X. \quad (4.27)$$

In practice, the convergence behavior of Alg. [4.1.1] as well as that of the SQP algorithm seems to be unchanged if at every iteration $i$ of Alg. [4.1.1] the already available value $\|z_{i-1}^{(\ell_{i-1})}\|_X$ is used in place of $\|z_i^{(\ell_i)}\|_X$, when applying criterion (4.27) (see Chapter [5]). This amounts to a simple stopping criterion that is nonintrusive with respect to the iterative augmented system solver.
Condition (4.13)

Assumption (4.12) of Thm. 4.1.6 motivates a sufficient condition for (4.13),

\[ \| \tilde{W}_k g_k \|_X \leq C_2 \| \tilde{W}_k^* g_k \|_X. \]

We obtain

\[
\begin{align*}
\| \tilde{W}_k g_k \|_X & \leq \| \tilde{W}_k g_k - \tilde{W}_k^* g_k \|_X + \| \tilde{W}_k^* g_k \|_X \\
& \leq 2 \| \tilde{W}_k - W_k \|_{L(X)} \| g_k \|_X + \| \tilde{W}_k^* g_k \|_X \\
& \leq 2(1 + \| S_i \|_2) (i + 1) C_1 \| \tilde{W}_k g_k \|_X \| g_k \|_X + \| \tilde{W}_k^* g_k \|_X,
\end{align*}
\]

thus

\[ \| \tilde{W}_k g_k \|_X \leq \frac{1}{1 - 2(1 + \| S_i \|_2) (i + 1) C_1} \| \tilde{W}_k^* g_k \|_X. \]

Unfortunately, as \( C_1 < 1 / (2(1 + \| S_i \|_2) (i + 1)) \) is implicit in the last inequality, and the previous section implies \( C_1 = \| G_k^{-1} M_k \|_{L(X \times Y)} \), we would need to change the main termination criterion (4.27) for the augmented system solver to

\[
\| e_i^{(\ell_i)} \|_{X \times Y} \leq \min \left\{ \frac{\| \tilde{W}_k g_k \|_X}{\| g_k \|_X}, \frac{\Delta_k}{\| g_k \|_X} \right\},
\]

\[
\min \left\{ \frac{\| z_i^{(\ell_i)} \|_X}{2(1 + \| S_i \|_2) (i + 1) \| G_k^{-1} M_k \|_{L(X \times Y)}}, \frac{1}{2(1 + \| S_i \|_2) (i + 1) \| G_k^{-1} M_k \|_{L(X \times Y)}} \right\}.
\]

This modification was not necessary for any of the numerical tests. We believe that condition (4.13) is implied in other convergence requirements and/or can be enforced in a more convenient fashion. Intuitively, as we approach the solution of the optimization problem, \( \tilde{W}_k g_k \) tends to zero, and the term \( 1 / (2(1 + \| S_i \|_2) (i + 1) \| G_k^{-1} M_k \|_{L(X \times Y)}) \) above can be omitted.
Explicitly ensuring that
\[
\frac{\| \bar{W}_r - W_r \|_X}{\| \bar{W}_r \|_X} \leq C_3,
\]
is fairly easy. From (4.24), we get
\[
\frac{\| \bar{W}_r - W_r \|_X}{\| \bar{W}_r \|_X} = \frac{\| P^T G_k^{-1} M_k e_i \|_X}{\| z_i \|_X} \leq \| G_k^{-1} M_k \|_{L(X \times Y)} \| e_i \|_{X \times Y} \| z_i \|_X,
\]
therefore the only necessary requirement is that if the augmented system solve terminates at its \( \ell_i \)-th iteration, then the following must be satisfied,
\[
\| e_i^{(\ell_i)} \|_{X \times Y} \leq C_{\text{hess}} \| z_i^{(\ell_i)} \|_X,
\]
(4.28)
where \( C_{\text{hess}} > 0 \) is some constant independent of \( i \) and \( k \) (note that the boundedness of \( \{ \| G_k^{-1} M_k \|_{L(X \times Y)} \} \) has already been discussed). In practice, choosing \( C_{\text{hess}} \) to be a small constant (e.g. \( 10^{-3} \)) or tying it to the stopping tolerance of Alg. 4.1.1 proves very effective in improving the local convergence behavior of Alg. 4.1.1 (and consequently that of the SQP algorithm). As before, the already available value \( \| z_i^{(\ell_{i-1})} \|_X \) can be used in place of \( \| z_i^{(\ell_i)} \|_X \) when applying criterion (4.28). Additionally, stopping condition (4.28) can be combined with stopping conditions (4.26) and (4.27).

4.2 Quasi–Normal Step

The requirements (3.1)
\[
\| n_k \|_X \leq \kappa_1 \| c(x_k) \|_Y,
\]
and (3.2)
\[
\| c(x_k) \|_Y^2 - \| c_x(x_k) n_k + c(x_k) \|_Y^2 \geq \kappa_2 \| c(x_k) \|_Y \min \{ \kappa_3 \| c(x_k) \|_Y, \Delta_k \},
\]
on the quasi-normal step are rather weak and can be satisfied using many algorithms for the minimization of a quadratic function subject to a spherical constraint. These algorithms can either be simple modifications of their “exact” counterparts (our dog-leg approach is a good candidate), or explicitly inexact schemes, usually based on Krylov subspace methods (for example, see Golub and von Matt [45] and Sorensen [70]).

Here we state a theoretical result related to an inexact variant of the dogleg algorithm 2.1.3.

**Lemma 4.2.1.** Let the inexact quasi-normal step $n_k$ be computed using a modification of Algorithm 2.1.3 in which the inexact Newton step $\hat{n}^N_k$ is obtained by solving the system

$$
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
y
\end{pmatrix}
= \begin{pmatrix}
r_1^k \\
c(x_k) + r_2^k
\end{pmatrix},
$$

(4.29)

and setting $\hat{n}^N_k = c_x(x_k)^*y$, with $r_1^k \in X$, $\|r_1^k\|_X \leq \epsilon$ and $r_2^k \in Y$, $\|r_2^k\|_Y \leq \epsilon$ for some $\epsilon > 0$ independent of $k$. Also assume that for every $k$

$$
\|c_x(x_k)r_1^k - r_2^k\|_Y \leq \|c_x(x_k)n_k^{cp} + c(x_k)\|_Y. 
$$

(4.30)

Then $n_k$ satisfies

$$
\|n_k\|_X \leq \kappa_1\|c(x_k)\|_Y
$$

and

$$
\|c(x_k)\|_Y^2 - \|c_x(x_k)n_k + c(x_k)\|_Y^2 \geq \|c(x_k)\|_Y^2 - \|c_x(x_k)n_k^{cp} + c(x_k)\|_Y^2,
$$

thereby satisfying the requirements (3.1) and (3.2).

**Proof.** It is easily shown from (4.29) that the inexact Newton step $\hat{n}^N_k$ is given by

$$
\hat{n}^N_k = n_k^N + c_x(x_k)^* (c_x(x_k)c_x(x_k)^*)^{-1} (c_x(x_k)r_1^k - r_2^k)
$$

$$
= -c_x(x_k)^* (c_x(x_k)c_x(x_k)^*)^{-1} c(x_k) + c_x(x_k)^* (c_x(x_k)c_x(x_k)^*)^{-1} (c_x(x_k)r_1^k - r_2^k).
$$
The inexact quasi-normal step \( n_k \) then satisfies

\[
n_k = n_k^{cp} + \hat{\theta}_N^k (n_k - n_k^{cp}),
\]

where \( 0 \leq \hat{\theta}_N^k \leq 1 \), \( n_k^{cp} = \alpha_k^{cp} c_x(x_k)^* c(x_k) \), and \( |\alpha_k^{cp}| \leq \alpha \), where \( \alpha \) is independent of \( k \) and \( \|c(x_k)\|_Y \). This is a trivial consequence of (2.19) and general boundedness assumptions (A4), which imply that there exist positive constants \( \nu_1 \) and \( \nu_2 \) such that

\[
\| (c_x(x)c_x(x)^*)^{-1} \|_{L(Y)} \leq \nu_1 \quad \text{and} \quad \|c_x(x)\|_{L(X,Y)} = \|c_x(x)^*\|_{L(Y,X)} \leq \nu_2 \quad \text{for all} \ x \in \Omega.
\]

Consequently, for any \( k \)

\[
\|n_k\| \leq 2\|n_k^{cp}\|_X + \|\hat{\theta}_N^k\| \leq 2\alpha \nu_2\|c(x_k)\|_Y + \nu_2 \nu_1\|c_x(x_k)n_k^{cp} + c(x_k)\|_Y
\]

\[
\leq 2\alpha \nu_2\|c(x_k)\|_Y + \nu_2 \nu_1\|c_x(x_k)n_k^{cp} + c(x_k)\|_Y + \nu_2 \nu_1\| (1 - \hat{\theta}_N^k) (c_x(x_k)n_k^{cp} + c(x_k)) + \theta_N^k (c_x(x_k)r_k - r_k^2) \|_Y
\]

\[
= \kappa_1\|c(x_k)\|_Y,
\]

where \( \kappa_1 = \nu_2 (2\alpha + \nu_1 + \nu_1 (2\nu_2 \alpha \nu_2 + 1)) \).

The second part of the proof is only slightly more involved. Using the fact that \( c_x(x_k)n_k^N + c(x_k) = 0 \), after a simple calculation we obtain

\[
\|c(x_k)\|_Y^2 - \|c_x(x_k)n_k + c(x_k)\|_Y^2
\]

\[
= \|c(x_k)\|_Y^2 - \|((1 - \theta_N^k) (c_x(x_k)n_k^{cp} + c(x_k)) + \theta_N^k (c_x(x_k)r_k^1 - r_k^2)) \|_Y^2.
\]

(4.31)

In order to simplify the notation we introduce

\[
v_k^{cp} = c_x(x_k)n_k^{cp} + c(x_k), \quad v_k^r = c_x(x_k)r_k^1 - r_k^2,
\]

where from the assumptions of the lemma \( \|v_k^{cp}\|_Y^2 \leq \|v_k^{cp}\|_Y^2 \). After expanding the last
term of the expression (4.31), we have

\[
\|c(x_k)\|_Y^2 - \|c_x(x_k)n_{k} + c(x_k)\|_Y^2
= \|c(x_k)\|_Y^2 - \|c_x(x_k)n_{k}^{cp} + c(x_k)\|_Y^2 +
\left\{2\theta_N^2 v_{k}^{cp} - \theta_N^2 \|v_{k}^{cp}\|_Y^2 - \theta_N^2 \|v_{k}^{r}\|_Y^2 - 2\theta_N^2 \langle v_{k}^{cp}, v_{k}^{r}\rangle_Y + 2\theta_N^2 \langle v_{k}^{cp}, v_{k}^{r}\rangle_Y\right\}
\geq \|c(x_k)\|_Y^2 - \|c_x(x_k)n_{k}^{cp} + c(x_k)\|_Y^2 + \left\{\theta_N^2 \|v_{k}^{cp}\|_Y^2 - 2\theta_N^2 \langle v_{k}^{cp}, v_{k}^{r}\rangle_Y +
\tilde{\theta}_N^2 \|v_{k}^{r}\|_Y^2 - \tilde{\theta}_N^2 \|v_{k}^{cp}\|_Y^2 + 2\tilde{\theta}_N^2 \langle v_{k}^{cp}, v_{k}^{r}\rangle_Y - \theta_N^2 \|v_{k}^{r}\|_Y^2\right\}
\geq \|c(x_k)\|_Y^2 - \|c_x(x_k)n_{k}^{cp} + c(x_k)\|_Y^2 + \left\{\tilde{\theta}_N^2 \|v_{k}^{r}\|_Y^2 - \theta_N^2 \|v_{k}^{cp} - v_{k}^{r}\|_Y^2 - \tilde{\theta}_N^2 \|v_{k}^{cp} - v_{k}^{r}\|_Y^2\right\}
\geq \|c(x_k)\|_Y^2 - \|c_x(x_k)n_{k}^{cp} + c(x_k)\|_Y^2,
\]

where we have used \(\tilde{\theta}_N^2 \geq \theta_N^2\).

It should be noted that the requirement (4.30) can be easily implemented as a stopping criterion for an iterative linear system solver.

### 4.3 Lagrange Multipliers

The computation of the Lagrange multipliers involves inexact linear system solves. Here we assume that the augmented system (2.12) in Section 2.1.2 is solved iteratively. On the other hand, global convergence proofs for SQP methods are merely based on the boundedness of Lagrange multipliers, as stated in assumption (A7). From the computational point of view, this requirement is quite easily imposed even when inexact linear system solvers are used. The key is usually in the remaining assumptions (A1)-(A6). For example, we can formulate the following lemma.

**Lemma 4.3.1.** Let the sequence \(\lambda_k\) of inexact Lagrange multipliers be generated by
solving the linear system
\[
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
\lambda_k
\end{pmatrix}
= \begin{pmatrix}
-\nabla_x f(x_k) + r^1_k \\
r^2_k
\end{pmatrix},
\]
with $r^1_k \in X$, $\|r^1_k\|_X \leq \epsilon$ and $r^2_k \in Y$, $\|r^2_k\|_Y \leq \epsilon$ for some $\epsilon > 0$ independent of $k$.
Then the sequence $\{\lambda_k\}$ of Lagrange multipliers is bounded.

Proof. From (4.32) we directly obtain
\[
\lambda_k = (c_x(x_k)c_x(x_k)^*)^{-1} \left(-c_x(x_k)\nabla_x f(x_k) + c_x(x_k)r^1_k - r^2_k\right)
= -(c_x(x_k)c_x(x_k)^*)^{-1} c_x(x_k) \nabla_x f(x_k) + (c_x(x_k)c_x(x_k)^*)^{-1} (c_x(x_k)r^1_k - r^2_k).
\]
General assumptions (A4) give uniform boundedness of the quantities $\|c_x(x)\|_{L(X,Y)}$, $\|(c_x(x)c_x(x)^*)^{-1}\|_{L(Y)}$, and $\|\nabla_x f(x)\|_X$ for all $x$. The claim follows from elementary norm inequalities and the boundedness of the residual norms $\|r^1_k\|_X$ and $\|r^2_k\|_Y$.

4.4 Balancing the Tangential and the Quasi–Normal Step

Having implicitly obtained a step $w_k$ (through the computed step $t_k$), we focus on the computation of the effective tangential step $\tilde{t}_k$.

We should immediately note that the computed step $t_k = \tilde{W}_k w_k$ need not equal the effective step $\tilde{t}_k$. In fact, as $t_k$ (and clearly $w_k$) can have a significant component outside of the range of $W_k$, it makes sense to perform an additional projection of the computed step $t_k$ (and thus implicitly project $w_k$) onto the null space of the constraints, in order to reduce potential loss of linear feasibility.

With exact linear system solves, this would amount to computing the effective tangential step $\tilde{t}_k = W_k t_k = W_k W_k w_k = W_k w_k = t_k$, i.e. we take the step $t_k$ as the effective tangential step. No additional projection is necessary. With inexactness, we compute
\[
\tilde{t}_k = \tilde{W}_k t_k = \tilde{W}_k \tilde{W}_k w_k,
\]
i.e. an additional projection must be performed, with sufficient accuracy so that the component of the effective tangential step outside of $\text{Null}(c_x(x_k))$ does not significantly violate the progress made by the quasi-normal step.

In particular, we have to make sure that the general condition (3.11),

$$|r_{\text{pred}}(r^t_k; \rho_k)| \leq \eta_{0\text{pred}}(n_k, w_k; \rho_k),$$

is satisfied. This is done by performing the aforementioned projection $\tilde{t}_k = W_k t_k$ with sufficient accuracy. The requirement (3.11) can always be satisfied, as $W_k \rightarrow W_k$ implies $\|r^t_k\|_Y \rightarrow 0$. We also note that $w_k$ is not required explicitly to compute the quantity $\text{pred}(n_k, w_k; \rho_k)$, as $t_k$ is readily available, and $\tilde{q}_k(w_k) = q_k(t_k)$.

The practical mechanisms addressing the remaining general conditions imposed on the effective tangential step are discussed below.

**Condition (3.14)**

In order to give a practical mechanism for satisfying

$$\|w_k\|_X \leq \xi_5 \|s^x_k\|_X,$$

we require that

$$\|w_k\|_X \leq C_5 \|\tilde{t}_k\|_X,$$

where $C_5 > 0$ is independent of $k$. We believe that this is a very reasonable assumption, as without inexactness in linear system solves $\tilde{t}_k = t_k = w_k$.

Consequently, we need to ensure that

$$\|\tilde{t}_k\|_X \leq C_6 \|s^x_k\|_X. \quad (4.34)$$

where we recall that the trial step $s^x_k$ is given by $s^x_k = n_k + \tilde{t}_k$. Our first considerations concern the quasi-normal step $n_k$. If the inexact quasi-normal step is computed
using, for example, the procedure given in Section 4.2, then

\[ n_k \in \text{Range}(c_x(x_k)^*) , \]

as \( n_k \) is a linear combination of \( \tilde{n}_k^N \) and \( n_k^{cp} \), and \( \tilde{n}_k^N, n_k^{cp} \in \text{Range}(c_x(x_k)^*) \). This means that the inexact quasi-normal step is in fact normal to the null space of the linearized constraints, if computed as in Section 4.2. This implies that we can control the angle between \( n_k \) and \( \tilde{t}_k \) by changing the accuracy of the projection \( \tilde{t}_k = W_k t_k \).

We present two arguments. The first argument is merely stated in order to better illustrate the nature of the requirement (4.34); it does not tell us how to enforce (4.34). Let \( 0 < \varrho < 1 \) (for instance \( \varrho = 0.1 \)), and assume that

\[ \| \tilde{t}_k \|_X \leq (1 - \varrho) \| n_k \|_X \quad \text{or} \quad \| \tilde{t}_k \|_X \geq (1 + \varrho) \| n_k \|_X . \quad (4.35) \]

The first inequality implies \( \| s_x^k \|_X \geq \varrho \| n_k \|_X \), which by another application of the first inequality means that \( \| s_x^k \|_X \geq \frac{\varrho}{1 - \varrho} \| \tilde{t}_k \|_X \), finally yielding

\[ \| \tilde{t}_k \|_X \leq \frac{1 - \varrho}{\varrho} \| s_x^k \|_X . \]

From the second inequality, we obtain \( \| n_k \|_X \leq \frac{1}{1 + \varrho} \| \tilde{t}_k \|_X \), which implies \( \| s_x^k \|_X \geq (1 - \frac{1}{1 + \varrho}) \| \tilde{t}_k \|_X \), and therefore

\[ \| \tilde{t}_k \|_X \leq \frac{1 + \varrho}{\varrho} \| s_x^k \|_X . \]

In summary, if (4.35) holds, the relation (4.34) is satisfied.

The second argument is of practical importance and covers all scenarios, in particular the one in which (4.35) is violated. If \( \tilde{t}_k \) is computed exactly, then \( n_k \perp \tilde{t}_k \), and thus \( \| s_x^k \|_X =\sqrt{\| n_k \|_X^2 + \| \tilde{t}_k \|_X^2} \). With inexactness, the angle \( \alpha \) between \( n_k \) and \( \tilde{t}_k \), see Figure 4.1, can be very small, causing \( \| s_x^k \|_X \) to shrink. In this case, we would like to ensure that the acute angle \( \alpha \) between \( n_k \) and \( \tilde{t}_k \) is greater than some fixed
value, in order to avoid excessively small trial steps $s_k^x$. Figure 4.1: Balancing the quasi–normal and the effective tangential step.

Let $0 < \alpha_0 < \frac{\pi}{2}$ be a fixed constant, independent of $k$ (for example $\alpha_0 = 0.1$). In Figure 4.1 let $r \perp \tilde{t}_k$ and let $n_k + z = r$. Without loss of generality, assume $\|z\|_X \leq \|\tilde{t}_k\|_X$. We have

$$\|z\|_X = \frac{1}{\tan \alpha} \|r\|_X, \quad \|\tilde{t}_k - z\|_X \leq \|s_k^x\|_X, \quad \text{and} \quad \|r\|_X \leq \|s_k^x\|_X,$$

therefore

$$\|\tilde{t}_k\|_X = \|z\|_X + \|\tilde{t}_k - z\|_X \leq \left( \frac{1}{\tan \alpha} + 1 \right) \|s_k^x\|_X.$$

In practice, we follow the following procedure:

If $\langle n_k, \tilde{t}_k \rangle < 0$ (acute angle $\alpha$), recompute the projection $\tilde{t}_k = W_k t_k$ with sufficient accuracy so that $\alpha > \alpha_0$ is satisfied. (4.36)
The condition $\alpha > \alpha_0$ can be used directly as a stopping criterion for the augmented system solve associated with the application of $W_k$.

**Conditions (3.12) and (3.13)**

The requirement (3.13) $\|\tilde{t}_k\|_X \leq \xi_4 \Delta_k$, follows from the definition $\tilde{t}_k = W_k t_k$, the standard algorithmic assumption $\|t_k\|_X \leq \Delta_k$ (see Step 1g of Alg. 3.6.2), and the boundedness of $W_k$, which can be ensured through a uniform bound on the norm of the residual in the augmented system solve and general assumptions (A4) and (A5).

The effective tangential step $\tilde{t}_k$ also needs to satisfy the condition (3.12) $\|\tilde{t}_k - W_k w_k\|_X \leq \xi_3 \Delta_k \|s_k\|_X$, which implies only minor modifications of the existing stopping criteria. Using the property $W_k W_k = W_k$, we note

$$
\|\tilde{t}_k - W_k w_k\|_X = \|W_k \tilde{W}_k w_k - W_k W_k w_k\|_X \\
= \|W_k \tilde{W}_k w_k - W_k \tilde{W}_k w_k + W_k \tilde{W}_k w_k - W_k \tilde{W}_k w_k\|_X \\
\leq \|W_k - W_k\| \|\tilde{W}_k w_k\|_X + \|W_k (\tilde{W}_k - W_k) w_k\|_X.
$$

The second term in the last expression is easily handled. By imposing the condition

$$
\|e_i^{(\ell_i)}\|_{X \times Y} \leq \Delta_k \|z_i^{(\ell_i)}\|_X,
$$

on the $\ell_i$-th iteration of the augmented system solve, we can ensure that

$$
\|\tilde{W}_k - W_k\|_{L(X)} \leq C_7 \Delta_k,
$$
for $C_7 > 0$ independent of $k$. We subsequently use the fact that $W_k$ is bounded and recall relation (3.14).

In order to analyze the first term, we note $\tilde{t}_k = W_k\tilde{W}_k w_k = W_k t_k$. The projection is applied by solving a preconditioned augmented system, thus $\tilde{t}_k$ satisfies

$$\tilde{t}_k = P^T G_k^{-1} (P t_k + M_k e_k),$$

where $e_k \in \mathcal{X} \times \mathcal{Y}$ is the corresponding preconditioned residual vector. Analogously to (4.24),

$$W_k t_k - W_k t_k = P^T G_k^{-1} M_k e_k.$$

Therefore, if we ensure that at the last iteration of the projection $\tilde{t}_k = W_k t_k$

$$\|e_k\|_{\mathcal{X} \times \mathcal{Y}} \leq \Delta_k \|n_k + \tilde{t}_k\|_{\mathcal{X}},$$

we obtain

$$\|(W_k - W_k)\tilde{W}_k w_k\|_{\mathcal{X}} \leq \|G_k^{-1} M_k\|_{L(\mathcal{X} \times \mathcal{Y})} \Delta_k \|s_k\|_{\mathcal{X}}.$$

Condition (4.38) can be satisfied directly by monitoring the preconditioned residual vector and the current iterate.

### 4.5 Statement of the Trust–Region SQP Algorithm with Inexact Augmented System Solves

This section focuses on the statement of a complete trust–region SQP algorithm that controls the accuracy in inexact augmented system solves.

We start by reminding the reader that the tangential step computation is based
on projections \( z_i = W_k(r_i) \), such that

\[
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z_i \\
y
\end{pmatrix} =
\begin{pmatrix}
r_i \\
0
\end{pmatrix} +
\begin{pmatrix}
e^1_i \\
e^2_i
\end{pmatrix}.
\]

Here we collect conditions discussed in Section 4.1.2. Let \( i \) be the running index of the CG iteration, let \( \ell_i \) be the running index of the linear solver iteration. We require the orthogonality conditions

\[
\tilde{Y}_i^* R_i \text{ is invertible \quad and \quad } \| S_i \|_2 < C_{\text{ortho}}. \quad (4.39)
\]

Conditions (4.25), (4.28), and (4.37) which must hold at the final linear solver iteration \( \ell_i \) for every CG iteration \( i \), can be combined into

\[
\| e^{(\ell_i)}_i \|_{X \times Y} \leq \min \left\{ \frac{\| z_0^{(\ell_i)} \|_X}{\| g_k \|_X}, \frac{\Delta_k}{\max\{1, \| g_k \|_X\}}, C_{\text{hess}} \right\} \| z_0^{(\ell_i)} \|_X. \quad (4.40)
\]

We note that the recycling of \( \| z_0^{(\ell_i-1)} \|_X \) is possible and performs well in practice. Finally, in the first iteration of Alg. 4.1.1, the fraction of Cauchy decrease condition (4.19) has to be satisfied, which can be guaranteed via (4.22).

For the following algorithm, we choose parameters \( i_{\text{max}} = 100, \ tol^{CG} = 10^{-2}, \ C_{\text{ortho}} = 10, \) and \( C_{\text{hess}} = 10^{-3}. \)

**Algorithm 4.5.1.** *(A practical full-space CG for the solution of (3.24))*

0. Let \( t_0 = 0 \in \mathcal{X} \). Let \( r_0 = g_k \). Set \( i_{\text{max}}, \ tol^{CG}, \ C_{\text{ortho}}, \) and \( C_{\text{hess}}. \)

1. For \( i = 0, 1, ..., i_{\text{max}} \)

   (a) Compute \( z_i = W_k(r_i) \) satisfying condition (4.40).

   If \( \| z_i \|_X \leq tol^{CG}, \) return \( t_i. \) Stop.

   (b) \( p_i = -z_i + \sum_{j=0}^{i-1} \frac{(z_i, H_k p_j)_X}{(p_j, H_k p_j)_X} p_j. \)

   (c) If \( \langle r_i, p_i \rangle_X \neq 0 \) and \( \langle p_i, H_k p_i \rangle_X \leq 0, \) compute \( \theta \) such that \( \text{sign}(\theta) = \text{sign}(- \langle r_i, p_i \rangle_X) \) and \( \| t_i + \theta p_i \|_X = \Delta_k - \| n_k \|_X, \) and return \( t_{i+1} = t_i + \theta p_i. \)

   If \( \langle r_i, p_i \rangle_X = 0 \) and \( \langle p_i, H_k p_i \rangle_X < 0, \) compute \( \theta \) such that \( \| t_i + \theta p_i \|_X = \Delta_k - \| n_k \|_X, \) and return \( t_{i+1} = t_i + \theta p_i. \)
We must then ensure that the effective tangential step $\tilde{t}_k$ satisfies (3.11), (4.36), and (4.38). Due to the trial–and–error nature of conditions (3.11) and (4.36), this step might require one or more additional projections.

We now have the necessary pieces to formulate the full trust-region SQP algorithm based on inexact linear system solves. The parameters used in Alg. 4.5.2, Alg. 3.4.1, and condition (3.11), for all numerical experiments in Chapter 5, are as follows: $\alpha_1 = 0.5$, $\Delta_0 = 10^2$, $\Delta_{\text{min}} = 10^{-8}$, $\Delta_{\text{max}} = 10^8$, $\eta_1 = 10^{-8}$, $\eta_0 = 0.5$, $\rho_{-1} = 1$, $\bar{\rho} = 10^{-8}$, and $\text{tol}^{\text{SQP}} = 10^{-6}$.

**Algorithm 4.5.2.** *(Trust-region SQP algorithm with inexact linear system solves.)*

1. Initialization. Choose initial point $x_0$, initial trust-region radius $\Delta_0$, constants $0 < \alpha_1, \eta_1 < 1$, $0 < \eta_0 < 1 - \eta_1$, $\rho_{-1} \geq 1$, and $\text{tol}^{\text{SQP}} > 0$. Set $\Delta_{\text{min}}, \Delta_{\text{max}}$ so that $0 < \Delta_{\text{min}} < \Delta_{\text{max}}$. Compute an initial Lagrange multiplier estimate $\lambda_0$.

2. For $k = 0, 1, 2, ...$
   
   (a) Convergence Check. If
   
   $$\|\nabla_x L(x_k, \lambda_k)\|_X < \text{tol}^{\text{SQP}} \quad \text{and} \quad \|c(x_k)\|_Y < \text{tol}^{\text{SQP}},$$
   
   then terminate.

   (b) Step Computation.
   
   i. Compute a quasi-normal step $n_k$ following Section 4.2
   
   ii. Compute a tentative tangential step $t_k$ via Alg. 4.5.1
(c) Acceptance Test.

i. Compute a new Lagrange multiplier estimate $\lambda_{k+1}$ based on $n_k + t_k$, following Section 4.3.

ii. Update penalty parameter $\rho_k$ via Alg. 3.4.1.

iii. Compute effective tangential step $\tilde{t}_k$ satisfying (3.11), (4.36), and (4.38).

iv. Compute trial step $s_k^r = n_k + \tilde{t}_k$.

v. Compute actual reduction $ared(s_k^r; \rho_k)$ and predicted reduction $pred(n_k, w; \rho_k)$, and their ratio $\theta_k = \frac{ared(s_k^r; \rho_k)}{pred(n_k, w; \rho_k)}$.

vi. If $\theta_k \geq \eta_1$, set $x_{k+1} = x_k + s_k^r$, and choose $\Delta_{k+1}$ such that

$$\max\{\Delta_{min}, \Delta_k\} \leq \Delta_{k+1} \leq \Delta_{max}.$$

Otherwise set $x_{k+1} = x_k$, reset $\lambda_{k+1} = \lambda_k$, and set

$$\Delta_{k+1} = \alpha_1 \max\{\|n_k\|_X, \|\tilde{t}_k\|_X\}.$$
Chapter 5

Numerical Examples

The purpose of this chapter is to test the performance of Alg 4.5.2 on two PDE–constrained optimization problems. The first involves optimal control of Burgers equation in one spatial dimension, and allows us to quickly assess the validity of the devised stopping criteria for iterative linear system solves. The second is a two–dimensional vorticity minimization problem, commonly used as a test case in the control of fluid flows in computational fluid dynamics, and involves Navier–Stokes equations.

5.1 Optimal Control of Burgers Equation in One Dimension

The first set of numerical experiments focuses on a simple optimal control problem involving the steady–state Burgers equation in one dimension. The main goal here is to test the performance of mechanisms for inexactness control in the tangential step computation, on a numerical example that can be modeled rather quickly.

In Section 5.1.1 we discuss a generic problem formulation. Section 5.1.2 describes a problem discretization with piecewise linear finite elements and considers the quantities that need to be computed in order to apply our SQP algorithm. The precise setup of the problem and corresponding numerical results are given in Section 5.1.3.
5.1.1 Problem Formulation

We consider an optimal control problem involving the steady–state Burgers equation in one dimension. In the case of distributed controls the problem is given by

$$
\min \frac{1}{2} \int_0^1 (u(x) - u_d(x))^2 dx + \frac{\alpha}{2} \int_0^1 g^2(x) dx
$$

subject to

$$
-\nu u_{xx}(x) + u(x)u_x(x) = f(x) + g(x) \quad x \in (0, 1)
$$

$$
u u(0) = 0, \quad u(1) = 0.
$$

Here \( u \) and \( g \) play the role of states and controls, respectively, \( \alpha > 0 \) and \( \nu > 0 \) are given parameters, and \( u_d \) and \( f \) are given functions.

The state equations \( (5.1b)-(5.1c) \) are interpreted in the standard weak sense, which is introduced next. We define the state and control spaces

$$
U = \{ u \in H^1([0, 1]) : u(0) = 0 \text{ and } u(1) = 0 \}, \quad G = L^2([0, 1]),
$$

respectively, and the space of test functions \( V = U \). The weak form of the state equations is given by

$$
a(u, v) + b(u; u, v) - \langle g, v \rangle = \langle f, v \rangle \quad \forall v \in V,
$$
where

\[
\begin{align*}
    a(u, v) &= \int_0^1 \nu \frac{d}{dx}u(x)\frac{d}{dx}v(x)\,dx, \\
    b(u; v, w) &= \int_0^1 u(x)\frac{d}{dx}v(x)w(x)\,dx, \\
    \langle g, v \rangle &= \int_0^1 g(x)v(x)\,dx, \\
    \langle f, v \rangle &= \int_\Omega f(x)v(x)\,dx.
\end{align*}
\] (5.4, 5.5, 5.6, 5.7)

The weak form of the optimal control problem can be stated as follows,

\[
\begin{align*}
    \min & \quad \frac{1}{2} \int_0^1 (u(x) - u_d(x))^2\,dx + \frac{\alpha}{2} \int_0^1 g^2(x)\,dx \\
    \text{subject to} & \quad a(u, v) + b(u; u, v) - \langle g, v \rangle = \langle f, v \rangle \quad \forall v \in V.
\end{align*}
\] (5.8a, 5.8b)

The Lagrangian associated with the optimal control problem is given by

\[
L(u, g, \lambda) = \frac{1}{2} \int_0^1 (u(x) - u_d(x))^2\,dx + \frac{\alpha}{2} \int_0^1 g^2(x)\,dx + a(u, \lambda) + b(u; u, \lambda) - \langle g, \lambda \rangle - \langle f, \lambda \rangle.
\] (5.9)

If \( u \in U \) and \( g \in G \) solve (5.1), then there exists a Lagrange multiplier \( \lambda \in U \) such that the following equations are satisfied:

\[
\begin{align*}
    a(v, \lambda) + b(v; u, \lambda) + b(u; v, \lambda) &= -\int_0^1 (u(x) - u_d(x))v(x)\,dx \quad \forall v \in V, \\
    \alpha g - \lambda &= 0 \quad \text{in } \Omega, \\
    a(u, v) + b(u; u, v) - \langle g, v \rangle &= \langle f, v \rangle \quad \forall v \in V.
\end{align*}
\]
5.1.2 Problem Discretization

We approximate $u$ and $g$ by functions of the form

$$u_h(x) = \sum_{j=1}^{n_u} u_j \varphi_j(x)$$

(5.11)

and

$$g_h(x) = \sum_{j=1}^{n_g} u_j \psi_j(x)$$

(5.12)

where $\varphi_j(0) = \varphi_j(1) = 1$, $j = 1, ..., n_u$, and $\psi_j(0) = \psi_j(1) = 1$, $j = 1, ..., n_g$. We set

$$\vec{u} = (u_1, \ldots, u_{n_u})^T \quad \text{and} \quad \vec{g} = (g_1, \ldots, g_{n_g})^T.$$

If we insert these approximations into (5.1b), we obtain

$$A \vec{u} + N(\vec{u}) + B \vec{g} = \vec{f},$$

(5.13)

where $A \in \mathbb{R}^{n_u \times n_u}$, $B \in \mathbb{R}^{n_u \times n_g}$, $\vec{f} \in \mathbb{R}^{n_u}$, and $N(\vec{u}) \in \mathbb{R}^{n_u}$ are matrices or vectors given by

$$A_{ij} = \nu \int_0^1 \frac{d}{dx} \varphi_j(x) \frac{d}{dx} \varphi_i(x) dx,$$

$$B_{ij} = -\int_0^1 \psi_j(x) \varphi_i(x) dx,$$

$$(N(\vec{u}))_i = \sum_{j=1}^{n_u} \sum_{k=1}^{n_g} \int_0^1 \varphi_j(x) \varphi_k(x) \varphi_i(x) dx,$$

$$f_i = \int_0^1 f(x) \varphi_i(x) dx.$$

If we additionally define the matrices $M \in \mathbb{R}^{n_u \times n_u}$, $Q \in \mathbb{R}^{n_g \times n_g}$,

$$M_{ij} = \int_0^1 \varphi_j(x) \varphi_i(x) dx, \quad Q_{ij} = \int_0^1 \psi_j(x) \psi_i(x) dx,$$
we obtain the discretized optimal control problem

\[
\min_{\vec{u}, \vec{g}} \quad F(\vec{u}, \vec{g}) \equiv \frac{1}{2} (\vec{u} - \vec{u}_d)^T \mathbf{M} (\vec{u} - \vec{u}_d) + \frac{\alpha}{2} \vec{g}^T \mathbf{Q} \vec{g}
\]  
(5.14a)

s.t. \quad \mathbf{A} \vec{u} + \mathbf{N}(\vec{u}) + \mathbf{B} \vec{g} = \vec{f}. 
(5.14b)

For our numerical implementation, we use piecewise linear functions on equidistant subintervals, i.e., we subdivide \([0, 1]\) into \(n_x\) subintervals of length \(h = 1/n_x\) and define

\[
\varphi_i(x) = \begin{cases} 
  h^{-1}(x - (i - 1)h) & x \in [(i - 1)h, ih] \cap [0, 1], \\
  h^{-1}(-x + (i + 1)h) & x \in [ih, (i + 1)h] \cap [0, 1], \\
  0 & \text{else,}
\end{cases}
\]

where \(i = 1, \ldots, n_u = n_x - 1\), and

\[
\psi_i(x) = \begin{cases} 
  h^{-1}(x - (i - 2)h) & x \in [(i - 2)h, (i - 1)h] \cap [0, 1], \\
  h^{-1}(-x + ih) & x \in [(i - 1)h, ih] \cap [0, 1], \\
  0 & \text{else,}
\end{cases}
\]

where \(i = 1, \ldots, n_g = n_x + 1\). This choice yields

\[
\mathbf{M} = \frac{h}{6} \begin{pmatrix} 
  4 & 1 & \cdots & \cdot \\
  1 & 4 & 1 & \cdots \\
  \vdots & \vdots & \ddots & \cdot \\
  1 & 4 & 1 & \cdots \\
  1 & 4 & \cdots & \cdot
\end{pmatrix} \in \mathbb{R}^{n_u \times n_u}, \quad \mathbf{A} = \frac{\nu}{h} \begin{pmatrix} 
  2 & -1 & \cdots & \cdot \\
  -1 & 2 & -1 & \cdots \\
  \vdots & \vdots & \ddots & \cdot \\
  -1 & 2 & -1 & \cdots \\
  -1 & 2 & \cdots & \cdot
\end{pmatrix} \in \mathbb{R}^{n_u \times n_u},
\]
\[
B = \frac{h}{6} \begin{pmatrix}
1 & 4 & 1 \\
1 & 4 & 1 \\
\ddots & \ddots & \ddots \\
1 & 4 & 1 \\
1 & 4 & 1 \\
\end{pmatrix} \in \mathbb{R}^{n_u \times n_g},
Q = \frac{h}{6} \begin{pmatrix}
2 & 1 \\
1 & 4 & 1 \\
\ddots & \ddots & \ddots \\
1 & 4 & 1 \\
1 & 2 \\
\end{pmatrix} \in \mathbb{R}^{n_g \times n_g},
\]
and
\[
N(\vec{u}) = \frac{1}{6} \begin{pmatrix}
u_1u_2 + u_2^2 \\
-u_1^2 - u_1u_2 + u_2u_3 + u_3^2 \\
\vdots \\
-u_{i-1}^2 - u_{i-1}u_i + u_iu_{i+1} + u_{i+1}^2 \\
\vdots \\
-u_{n_u-2}^2 - u_{n_u-2}u_{n_u-1} + u_{n_u-1}u_{n_u} + u_{n_u}^2 \\
-u_{n_u-1}^2 - u_{n_u-1}u_{n_u} \\
\end{pmatrix} \in \mathbb{R}^{n_u}.
\]

The matrix \(N'(\vec{y}) \in \mathbb{R}^{n_u \times n_u}\) is shown in Figure 5.1. For the approximation of the integral arising in the definition of \(\vec{f}\), the composite trapezoidal rule yields
\[
\vec{f}_i = hf(ih),
\]
for \(i = 1, \ldots, n_u\).

For a complete implementation of this example within our SQP framework, we need to compute several additional quantities. The discretized Lagrangian functional is given by
\[
\mathcal{L}(\vec{u}, \vec{g}, \vec{\lambda}) = \frac{1}{2}(\vec{u} - \vec{u}_d)^T M(\vec{u} - \vec{u}_d) + \frac{\alpha}{2} \vec{g}^T Q \vec{g} + \vec{\lambda}^T (A\vec{u} + N(\vec{u}) + B\vec{g} - \vec{f}),
\]
where \(\vec{\lambda} \in \mathbb{R}^{n_u}\) is the discrete representation of the Lagrange multiplier vector. The
Figure 5.1: The matrix $N'(\vec{u})$. 

\[
N'(\vec{u}) = \frac{1}{6}
\]

\[
\begin{pmatrix}
    u_2 & u_1 + 2u_2 & u_3 - u_1 & u_2 + 2u_3 & \ldots
    & -2u_1 - u_2 & u_1 - u_3 & u_2 + 2u_4 & \ldots
    & -2u_2 - u_3 & u_3 - u_5 & u_2 + 2u_6 & \ldots
    & \vdots & \vdots & \vdots & \vdots & \ddots
    & -2u_n - u_{n+1} & u_n - u_{n+2} & u_{n+1} + 2u_{n+2} & \ldots
    & -2u_{n+1} - u_{n+2} & u_{n+2} - u_{n+3} & u_{n+1} + 2u_{n+3} & \ldots
\end{pmatrix}
\]
gradient of the objective function is

\[
\nabla_{\vec{u}, \vec{g}} F = \begin{pmatrix}
M(\vec{u} - \vec{u}_d) \\
\alpha Q \vec{g}
\end{pmatrix},
\]

and the Jacobian of the constraints is

\[
J(\vec{u}, \vec{g}) = \begin{pmatrix} A + N'(\vec{u}) & B \end{pmatrix}.
\]

For the computation of the Hessian of the Lagrangian, by the trilinearity properties of \(b(u; v, w)\) we have for an arbitrary vector \(\vec{s}_u \in \mathbb{R}^{n_u}\)

\[
\nabla^2_{\vec{u}} (\vec{\lambda}^T N(\vec{u})) \vec{s}_u = N'(\vec{s}_u)^T \vec{\lambda},
\]

and so for \(\vec{s} = (\vec{s}_u, \vec{s}_g) \in \mathbb{R}^{n_u+n_g}\)

\[
(\nabla^2_{\vec{u}, \vec{g}} \mathcal{L}) \vec{s} = \begin{pmatrix}
M \vec{s}_u + N'(\vec{s}_u)^T \vec{\lambda} \\
\alpha Q \vec{s}_g
\end{pmatrix}.
\]

The solution of augmented systems needed to compute the Lagrange multipliers, the quasi–normal step, and the tangential step involves the matrix

\[
\begin{pmatrix}
I_{n_u \times n_u} & 0_{n_u \times n_g} & (A + N'(\vec{u}))^T \\
0_{n_g \times n_u} & I_{n_g \times n_g} & B^T \\
A + N'(\vec{u}) & B & 0_{n_u \times n_u}
\end{pmatrix}.
\]

5.1.3 Numerical Results

For our numerical experiments, we choose the number of subintervals \(n_x = 100\), the viscosity parameter \(\nu = 10^{-2}\), the control penalty parameter \(\alpha = 10^{-5}\), the right–hand side forcing term \(f = 0\), and the desired state function \(u_d(x) = \sin(2\pi x)\). The code is implemented in Matlab.
As discussed previously, we focus on inexactness control in the computation of the tangential step only. The Lagrange multipliers and the quasi–normal step are computed using the algorithms of Chapter 2 with sparse direct linear system solves. The convergence conditions for the quasi–normal step are thus automatically satisfied.

The tangential step is computed using Alg. 4.5.1 with the option to turn inexactness control on or off. Our goal is to compare the two approaches. In particular, with inexactness control on, the full set of termination criteria and the step acceptance module properly modified for inexact linear system solves are used. The stopping tolerance for the first augmented system solve is computed based on the norm of the true projected residual, for all others, the norm of the projected residual from the previous iteration of Alg 4.5.1 is recycled. With inexactness control off, a fixed preset absolute stopping tolerance is used for all augmented system solves, along with a standard step acceptance scheme, i.e. without the additional null space projection and the \( r_{pred} \) modification. In both cases, we have noticed that the full reorthogonalization procedure in Step 1b of Alg 4.5.1 was necessary. If omitted, the CG scheme tends to diverge after only a few iterations.

For augmented system solves, we use GMRES (even though our systems are symmetric), as it was readily available. Any iterative solver that can handle symmetric indefinite systems can be used. It is well known that KKT (and augmented) systems arising from optimal control problems are notoriously ill-conditioned, and usually require \( O(n) \) linear solver iterations for convergence, where \( n \) is the size of the system. To accelerate convergence of GMRES, we use an incomplete LU preconditioner with a drop tolerance of \( 5 \times 10^{-3} \).

The first experiment examines the convergence of the SQP algorithm 4.5.2 with inexactness control in the tangential step. Figure 5.2 shows absolute GMRES stopping tolerances for the entire SQP run, for every CG iteration. For every SQP iteration, the symbol □ denotes the controlled tolerance in the first CG iteration. The symbol * denotes the controlled tolerances in all other CG iterations. We make the distinc-
tion for two reasons: one is purely informative (so that one can distinguish between successive SQP iterations), the other is in the fundamental difference in how the first stopping tolerance is computed in Alg 4.5.1.

The desired SQP convergence tolerance ($10^{-6}$ for both the norm of the constraints and the norm of the gradient of the Lagrangian) is achieved in 16 SQP iterations, and 100 CG iterations. The total number of GMRES iterations in all tangential step computations (and the additional null space projections) is 1910. Figure 5.2 shows that when the SQP iterate is far from the optimum, fairly loose GMRES stopping tolerances are allowed, roughly on the order of $10^{-6}$. As the SQP algorithm progresses toward a KKT point, absolute GMRES stopping tolerances are decreased, to approximately $10^{-12}$. This behavior is clearly very encouraging.

In order to properly gauge the computational cost of the GMRES solves with controlled tolerances, we consider Figure 5.3 which depicts relative stopping tolerances (with respect to the size of the right-hand side vectors in augmented system solves). What is remarkable about Figure 5.3 is that the required relative tolerances are mostly on the order of $10^{-4}$ to $10^{-5}$, and never below $10^{-7}$, which is only slightly smaller than the required SQP stopping tolerances.

The second set of experiments aims to contrast the inexactness control approach
with the current state of the art, which amounts to choosing a fixed linear solver tolerance that “works well”, purely by trial and error. We start with very small stopping tolerances (we could go as low as $5 \times 10^{-16}$ while still converging within the chosen maximum number of GMRES iterations), and raise them by one or two orders of magnitude. The results are presented in Table 5.1.

<table>
<thead>
<tr>
<th></th>
<th>inx. ctrl</th>
<th>5e-16</th>
<th>1e-15</th>
<th>1e-14</th>
<th>1e-13</th>
<th>1e-12</th>
<th>1e-10</th>
<th>1e-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>converges</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>GMRES iter’s</td>
<td>1910</td>
<td>2898</td>
<td>2629</td>
<td>2073</td>
<td>1978</td>
<td>3253</td>
<td>5563</td>
<td>&gt;10000</td>
</tr>
<tr>
<td>CG iter’s</td>
<td>100</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>101</td>
<td>199</td>
<td>389</td>
<td>&gt;500</td>
</tr>
<tr>
<td>SQP iter’s</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>20</td>
<td>37</td>
<td>&gt;50</td>
</tr>
</tbody>
</table>

**Table 5.1:** Inexactness control vs. fixed stopping tolerances – Burgers.

The first observation is that the inexactness control approach seems to recover similar local convergence behavior as high-precision fixed tolerance cases, see Tables 5.2 and 5.3. This is somewhat unexpected, as the devised stopping tolerances for linear solvers are solely based on global SQP convergence criteria. Second, the inexactness control approach always outperforms fixed tolerance runs in terms of the number of required GMRES iterations. Third, who could have guessed that a fixed
stopping tolerance of $10^{-13}$ would be our best fixed-tolerance option, and that it would yield the same results as $10^{-12}$ in only half the number of GMRES iterations? Finally, how could we have known that a fixed GMRES stopping tolerance of $10^{-8}$ (which is still rather small) would yield no convergence to the desired SQP stopping tolerances?

In summary, the inexactness control strategy performs quite well on this example. Inexactness control tests indicate a globally convergent algorithm, whose local convergence behavior rivals that of an exact scheme. We would also like to note that the incomplete LU preconditioner performed almost too well on this example, with most GMRES solves ending in less than 20 iterations, and the gap between an absolute residual of $10^{-6}$ and that of $10^{-12}$ being closed by only three or four iterations. For a typical GMRES convergence plot, see Figure 5.4. For more complicated problems, and less efficient preconditioners, we expect our inexactness control scheme to outperform the best fixed-tolerance guess by a significantly wider margin.

![Figure 5.4: Performance of preconditioned GMRES – Burgers.](image)
<table>
<thead>
<tr>
<th>iter</th>
<th>$f(x_k)$</th>
<th>$|c(x_k)|$</th>
<th>$|\nabla_x L_k|$</th>
<th>$\Delta_k$</th>
<th>$n_k$</th>
<th>$t_k$</th>
<th>CG iter</th>
<th>accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.43174e-01</td>
<td>1.480582e+00</td>
<td>6.064963e-02</td>
<td>1.000000e+02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7.43174e-01</td>
<td>1.480582e+00</td>
<td>6.064963e-02</td>
<td>5.000000e+01</td>
<td>3.998102e+00</td>
<td>1.000000e+02</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>7.43174e-01</td>
<td>1.480582e+00</td>
<td>6.064963e-02</td>
<td>2.500002e+01</td>
<td>3.998102e+00</td>
<td>5.000004e+01</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>1.80268e-01</td>
<td>1.044880e+00</td>
<td>5.137368e-02</td>
<td>5.063544e+01</td>
<td>3.998102e+00</td>
<td>2.500004e+01</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>1.80268e-01</td>
<td>1.044880e+00</td>
<td>5.137368e-02</td>
<td>2.531772e+01</td>
<td>9.999874e+00</td>
<td>5.063545e+01</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>1.80268e-01</td>
<td>1.044880e+00</td>
<td>5.137368e-02</td>
<td>1.265887e+01</td>
<td>9.999874e+00</td>
<td>2.531773e+01</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>1.80268e-01</td>
<td>1.044880e+00</td>
<td>5.137368e-02</td>
<td>6.329431e+00</td>
<td>9.999874e+00</td>
<td>1.265886e+01</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>5.95319e-02</td>
<td>5.060074e-01</td>
<td>1.848318e-02</td>
<td>1.621114e+01</td>
<td>5.063545e+00</td>
<td>6.329434e+00</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>1.56000e-02</td>
<td>1.025080e-01</td>
<td>6.250593e-03</td>
<td>1.271603e+02</td>
<td>8.197468e+00</td>
<td>1.621108e+01</td>
<td>5</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>1.56000e-02</td>
<td>1.025080e-01</td>
<td>6.250593e-03</td>
<td>6.358013e+01</td>
<td>6.953706e-01</td>
<td>1.271603e+02</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>1.56000e-02</td>
<td>1.025080e-01</td>
<td>6.250593e-03</td>
<td>3.179006e+01</td>
<td>6.953706e-01</td>
<td>6.358012e+01</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>1.56000e-02</td>
<td>1.025080e-01</td>
<td>6.250593e-03</td>
<td>1.589503e+01</td>
<td>6.953706e-01</td>
<td>3.179006e+01</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>12</td>
<td>5.64170e-03</td>
<td>1.117806e-01</td>
<td>1.151300e-03</td>
<td>1.589503e+01</td>
<td>6.953706e-01</td>
<td>1.589503e+01</td>
<td>5</td>
<td>Y</td>
</tr>
<tr>
<td>13</td>
<td>4.96091e-04</td>
<td>4.968187e-02</td>
<td>4.203988e-04</td>
<td>3.183471e+00</td>
<td>8.429511e-01</td>
<td>1.589502e+01</td>
<td>6</td>
<td>Y</td>
</tr>
<tr>
<td>14</td>
<td>3.70395e-05</td>
<td>1.755739e-03</td>
<td>4.355925e-05</td>
<td>4.325318e+01</td>
<td>5.983391e-01</td>
<td>6.149971e+00</td>
<td>10</td>
<td>Y</td>
</tr>
<tr>
<td>15</td>
<td>2.50740e-05</td>
<td>1.987861e-04</td>
<td>3.317000e-06</td>
<td>4.325318e+01</td>
<td>1.777556e-02</td>
<td>2.335972e+00</td>
<td>18</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 5.2: SQP convergence history with inexactness control – Burgers.
<table>
<thead>
<tr>
<th>iter</th>
<th>$f(x_k)$</th>
<th>$|c(x_k)|$</th>
<th>$|\nabla_x L_k|$</th>
<th>$\Delta_k$</th>
<th>$n_k$</th>
<th>$|t_k|$</th>
<th>CG iter</th>
<th>accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.43174e-01</td>
<td>1.480582e+00</td>
<td>6.064963e-02</td>
<td>1.000000e+02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7.43174e-01</td>
<td>1.480582e+00</td>
<td>6.064963e-02</td>
<td>5.000000e+01</td>
<td>3.998102e+00</td>
<td>1.000000e+02</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>7.43174e-01</td>
<td>1.480582e+00</td>
<td>6.064963e-02</td>
<td>2.500000e+01</td>
<td>3.998102e+00</td>
<td>5.000000e+01</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>1.80267e-01</td>
<td>1.044879e+00</td>
<td>5.137359e-02</td>
<td>5.063536e+01</td>
<td>3.998102e+00</td>
<td>2.500000e+01</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>1.80267e-01</td>
<td>1.044879e+00</td>
<td>5.137359e-02</td>
<td>2.531768e+01</td>
<td>9.999860e+00</td>
<td>5.063536e+01</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>1.80267e-01</td>
<td>1.044879e+00</td>
<td>5.137359e-02</td>
<td>1.265884e+01</td>
<td>9.999860e+00</td>
<td>2.531768e+01</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>1.80267e-01</td>
<td>1.044879e+00</td>
<td>5.137359e-02</td>
<td>6.329419e+00</td>
<td>9.999860e+00</td>
<td>1.265884e+01</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>5.95339e-02</td>
<td>5.060184e-01</td>
<td>1.848182e-02</td>
<td>1.621122e+00</td>
<td>5.063536e+00</td>
<td>6.329419e+00</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>1.55872e-02</td>
<td>1.023382e-01</td>
<td>6.272712e-03</td>
<td>1.271618e+02</td>
<td>8.197469e+00</td>
<td>1.621122e+01</td>
<td>5</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>1.55872e-02</td>
<td>1.023382e-01</td>
<td>6.272712e-03</td>
<td>6.358088e+01</td>
<td>6.962314e-01</td>
<td>1.271618e+02</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>1.55872e-02</td>
<td>1.023382e-01</td>
<td>6.272712e-03</td>
<td>3.179044e+01</td>
<td>6.962314e-01</td>
<td>6.358088e+01</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>1.55872e-02</td>
<td>1.023382e-01</td>
<td>6.272712e-03</td>
<td>1.589522e+01</td>
<td>6.962314e-01</td>
<td>3.179044e+01</td>
<td>5</td>
<td>N</td>
</tr>
<tr>
<td>12</td>
<td>5.66448e-03</td>
<td>1.121051e-01</td>
<td>1.158361e-03</td>
<td>1.589522e+01</td>
<td>6.962314e-01</td>
<td>1.589522e+01</td>
<td>5</td>
<td>Y</td>
</tr>
<tr>
<td>13</td>
<td>4.95763e-04</td>
<td>4.972136e-02</td>
<td>4.285732e-04</td>
<td>3.183540e+01</td>
<td>8.457309e-01</td>
<td>1.589522e+01</td>
<td>6</td>
<td>Y</td>
</tr>
<tr>
<td>14</td>
<td>3.96082e-05</td>
<td>1.899466e-03</td>
<td>3.582297e-05</td>
<td>4.091844e+01</td>
<td>5.897695e-01</td>
<td>5.814744e+00</td>
<td>9</td>
<td>Y</td>
</tr>
<tr>
<td>15</td>
<td>2.50772e-05</td>
<td>1.927458e-04</td>
<td>9.868444e-07</td>
<td>4.091844e+01</td>
<td>1.792903e-02</td>
<td>2.445594e+00</td>
<td>17</td>
<td>Y</td>
</tr>
<tr>
<td>16</td>
<td>2.50116e-05</td>
<td>4.725581e-07</td>
<td>1.265484e-08</td>
<td>4.091844e+01</td>
<td>6.361227e-04</td>
<td>1.143115e+00</td>
<td>31</td>
<td>Y</td>
</tr>
</tbody>
</table>

**Table 5.3:** SQP convergence history with exact solves – Burgers.
5.2 Optimal Control of Navier–Stokes Equations in Two Dimensions

In this section we consider an optimal control problem governed by the steady–state incompressible Navier–Stokes equations. Again, our goal is to test the performance of inexactness control in the tangential step computation. The structure of the problem is similar to that of optimal control of Burgers equation. It is, however, computationally much more demanding and from that point of view closer to a “real–world” application.

In Section 5.2.1 we discuss a generic problem formulation and set up a problem discretization while omitting low–level details. The precise setup of the test problem and corresponding numerical results are given in Section 5.2.2.

5.2.1 Problem Formulation

Let $\Omega \in \mathbb{R}^2$ be the channel with a backward facing step shown in Figure 5.5. We

![Figure 5.5: Geometry of the backward–facing step channel.](image-url)
consider an optimal control problem governed by the Navier–Stokes equations

\[-\nu \Delta u + (u \cdot \nabla)u + \nabla p = f \quad \text{in} \quad \Omega, \quad (5.15a)\]
\[\text{div} \ u = 0 \quad \text{in} \quad \Omega, \quad (5.15b)\]
\[(\nu \nabla u - p I) n = 0 \quad \text{on} \quad \Gamma_{\text{out}}, \quad (5.15c)\]
\[u = g \quad \text{on} \quad \Gamma_{\text{c}}, \quad (5.15d)\]
\[u = b \quad \text{on} \quad \Gamma_{\text{in}}, \quad (5.15e)\]
\[u = 0 \quad \text{on} \quad \partial \Omega \setminus (\Gamma_{\text{in}} \cup \Gamma_{c} \cup \Gamma_{\text{out}}). \quad (5.15f)\]

Later, \(g\) will be the control.

The Dirichlet boundary conditions \((5.15d,5.15e,5.15f)\) can be implemented through interpolation \([39]\), weakly through a Lagrange multiplier technique \([50]\), or via a penalty approach \([58, 57]\). We use interpolation to implement the Dirichlet boundary conditions \((5.15e,5.15f)\) with fixed data, and replace \((5.15d)\) by the penalized Neumann boundary condition

\[(\nu \nabla u - p I) n + \frac{1}{\delta} u = \frac{1}{\delta} g \quad \text{on} \quad \Gamma_{c}, \quad (5.16)\]

The state space is related to the function spaces

\[H^1 = H^1(\Omega) = \left\{ v_j \in L^2(\Omega) \mid \frac{\partial v_j}{\partial x_k} \in L^2(\Omega) \text{ for } j, k = 1, \ldots, N \right\},\]
\[H^1_D = H^1_D(\Omega) = \{ v \in H^1 \mid v = 0 \text{ on } \partial \Omega \setminus (\Gamma_{c} \cup \Gamma_{\text{out}}) \},\]

for the velocity vector, and

\[L^2 = L^2(\Omega)\]

for the pressure.

The weak form of the Navier-Stokes equations \((5.15a,5.15c,5.15e,5.15f,5.16)\) is given as follows. Find \(u \in H^1(\Omega)\) with \(u = b\) on \(\Gamma_{\text{in}}\) and \(u = 0\) on \(\partial \Omega \setminus (\Gamma_{c} \cup \Gamma_{\text{out}})\)
and \( p \in L^2(\Omega) \), such that

\[
\int_{\Omega} \nabla u : \nabla v \, dx + \int_{\Omega} (u \cdot \nabla)u \cdot v \, dx - \int_{\Omega} p \nabla \cdot v \, dx \\
- \int_{\Omega} q \nabla \cdot u \, dx + \frac{1}{\delta} \int_{\Gamma_c} v \cdot (u - g) \, dx \\
= \int_{\Omega} f \cdot v \, dx
\tag{5.17}
\]

for all \( v \in H^1_D(\Omega) \) and all \( q \in L^2(\Omega) \). With the definitions

\[
a(u, v) = \int_{\Omega} \nabla u : \nabla v \, dx, \tag{5.18}
\]

\[
b(u; v, w) = \int_{\Omega} ((u \cdot \nabla)v) \cdot w \, dx, \tag{5.19}
\]

\[
c(v, q) = -\int_{\Omega} \nabla \cdot v \, q \, dx, \tag{5.20}
\]

\[
\langle f, v \rangle = \int_{\Omega} f \cdot v \, dx, \tag{5.21}
\]

\[
\langle h, v \rangle_{\Gamma_c} = \int_{\Gamma_c} h \cdot v \, dx \tag{5.22}
\]

(5.17) is written as

\[
a(u, v) + b(u; u, v) + c(v, p) + c(u, q) + \frac{1}{\delta} \langle u, v \rangle_{\Gamma_c} - \frac{1}{\delta} \langle g, v \rangle_{\Gamma_c} = \langle f, v \rangle \tag{5.23}
\]

for all \( v \in H^1_D(\Omega) \) and all \( q \in L^2(\Omega) \).

We want to find a control \( g \) such that the vorticity in a region \( D = [1, 3] \times [0, 1/2] \) behind the step is reduced. We define

\[
J_1(u) = \frac{1}{2} \int_D (\partial_{x_1} u_2 - \partial_{x_2} u_1)^2 \, dx. \tag{5.24}
\]
For a given $\alpha > 0$ and $\delta > 0$ we consider the optimal control problem

$$\text{Minimize} \quad J_1(u) + \frac{\alpha}{2} \int_{\Gamma_c} |g|^2 dx,$$

(5.25a)

subject to

$$a(u, v) + b(u; u, v)$$

$$+ c(v, p) + c(u, q) + \frac{1}{\delta} \langle u, v \rangle_{\Gamma_c}$$

$$- \frac{1}{\delta} \langle g, v \rangle_{\Gamma_c} = \langle f, v \rangle \quad \forall v \in H^1_D(\Omega), q \in L^2(\Omega).$$

(5.25b)

The Lagrangian associated with the optimal control problem is given by

$$L(u, p, g, \lambda, \theta) = \quad J_1(u) + \frac{\alpha}{2} \int_{\Gamma_c} |g|^2 dx$$

$$+ a(u, \lambda) + b(u; u, \lambda)$$

$$+ c(\lambda, p) + c(u, \theta) + \frac{1}{\delta} \langle u, \lambda \rangle_{\Gamma_c}$$

$$- \frac{1}{\delta} \langle g, \lambda \rangle_{\Gamma_c} - \langle f, \lambda \rangle.$$ 

(5.26)

If $u \in H^1(\Omega)$ with $u = b$ on $\Gamma_{in}$ and $u = 0$ on $\partial \Omega \setminus (\Gamma_c \cup \Gamma_{out})$, $p \in L^2(\Omega)$ and $g \in L^2(\Gamma_c)$ solve (5.25), then there exist $\lambda \in H^1_D(\Omega)$ and $\theta \in L^2(\Omega)$ such that the following equations are satisfied:

$$a(v, \lambda) + b(v; u, \lambda) + b(u; v, \lambda)$$

$$+ c(\lambda, q) + c(v, \theta) + \frac{1}{\delta} \langle v, \lambda \rangle_{\Gamma_c} = -D_u J_1(u) v \quad \forall v \in H^1_D(\Omega), q \in L^2(\Omega),$$

(5.27a)

$$\alpha g - \frac{1}{\delta} \lambda = 0$$

on $\Gamma_c$.

(5.27b)

$$a(u, v) + b(u; u, v) + c(v, p) + c(u, q)$$

$$+ \frac{1}{\delta} \langle u, v \rangle_{\Gamma_c} - \frac{1}{\delta} \langle g, v \rangle_{\Gamma_c} = \langle f, v \rangle \quad \forall v \in H^1_D(\Omega), q \in L^2(\Omega).$$

(5.27c)
For the objective function (5.24), \( D_u J_1(u)v \) is given by

\[
D_u J_1(u)v = \int_D (\partial_{x_1} u_2 - \partial_{x_2} u_1)(\partial_{x_1} v_2 - \partial_{x_2} v_1) dx.
\]

We discretize the Navier–Stokes equations using Taylor–Hood finite elements \([44, 49, 51, 58]\). This discretization of the optimal control problem (5.25) leads to a nonlinear programming problem of the form

\[
\begin{align*}
\text{Minimize} & \quad \frac{1}{2} \bar{u}^T Q \bar{u} + \alpha \bar{g}^T R \bar{g}, \\
\text{subject to} & \quad \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{p} \end{pmatrix} + \begin{pmatrix} N(\bar{u}) \\ 0 \end{pmatrix} \\
& \quad + \begin{pmatrix} M_u \\ 0 \end{pmatrix} \bar{u} - \begin{pmatrix} M_g \\ 0 \end{pmatrix} \bar{g} = \begin{pmatrix} f \\ 0 \end{pmatrix}.
\end{align*}
\]

(5.28a)

(5.28b)

For brevity, we omit the details of the discretization. The quantities required for the application of our SQP algorithm can be computed as in Section 5.1.2, where the discrete representation (5.14) is replaced by (5.28).

### 5.2.2 Numerical Results

As mentioned previously, this test example considers vorticity minimization in fluid flows. The two–dimensional flow is set in a computational domain known as the backward–facing step channel, see Figure 5.5. This is a fairly common setup in the area of computational fluid dynamics.

At the inflow boundary \( \Gamma_{in} = \{0\} \times [0.5, 1] \) we assume that the flow is a parabolic function of \( x_2 \) only with \( u(x_2) = 8(x_2 - 0.5)(1 - x_2) \). The viscosity parameter is set to \( \nu = 5 \times 10^{-3} \). It is known that for such viscosity levels, the flow separates and recirculation occurs close to the corner region. The uncontrolled flow is depicted in Figure 5.7 with a more detailed snapshot of the corner region in Figure 5.8.
The control is applied on the boundary $\Gamma_c = \{1\} \times [0,0.5]$. We study the case in which suction/blowing is applied at $\Gamma_c$ with zero tangential velocity, i.e. the case of normal velocity control. This problem has been considered in more detail in [58]. The computational domain is divided into 352 triangles with a finer mesh across the area in which recirculation forms, see Figure 5.6. The parameter penalizing the magnitude of the velocity of the control fluid is set to $\alpha = 0.1$. The penalty parameter for the Dirichlet boundary conditions is taken to be $\delta = 10^{-5}$.

![Figure 5.6: The finite element grid for the backstep channel.](image)

For both uncontrolled and controlled flows, we have been able to reproduce the results of [58], which gives us confidence in our implementation of the Taylor–Hood finite element discretization for the solution of the Navier–Stokes equations, as well as the implementation of the quantities related to the objective function and the control terms. The controlled flow is given in Figure 5.9. The computed vorticity levels in the domain $D$ indicate that normal velocity control on the boundary $\Gamma_c$ can be very effective in eliminating recirculation.

We now focus on the true objective of this study, the inexactness control in the tangential step computation. The Lagrange multipliers and the quasi–normal step are computed using exact linear system solves. In a large–scale setting, exact solves can be replaced with high–accuracy iterative solves. This choice is particularly justified for the Navier–Stokes example, as the cost of the tangential step computation accounts for more than 85% of the total computational cost of the SQP algorithm. The experimental setup resembles that of Section 5.1. The drop tolerance for the incomplete LU preconditioner is lowered to $5 \times 10^{-5}$. 

The first experiment examines the convergence of the SQP algorithm with inexactness control. Figure 5.10 shows absolute GMRES stopping tolerances for the entire SQP run. The desired SQP convergence tolerance is achieved in 9 SQP iterations, and 162 CG iterations. The total number of GMRES iterations in all tangential step computations (and the additional null space projections) is 2672. As in the case of Burgers equation, when the SQP iterate is far from the optimum, fairly loose GMRES stopping tolerances are allowed, roughly on the order of $10^{-5}$. As the SQP algorithm moves toward a KKT point, absolute GMRES stopping tolerances are automatically
decreased to approximately $10^{-12}$. At the same time, the required relative tolerances, see Figure 5.11, are mostly on the order of $10^{-4}$ to $10^{-5}$ (and as low as $10^{-3}$), and only twice below $10^{-6}$.

The second set of experiments compares the inexactness control approach with several choices of fixed linear solver tolerances. We start with the stopping tolerance of $10^{-12}$ and increase it by one order of magnitude to $10^{-8}$. The results are presented in Table 5.4.
Figure 5.11: Relative GMRES stopping tolerances – Navier–Stokes.

Table 5.4: Inexactness control vs. fixed stopping tolerances – Navier–Stokes.

<table>
<thead>
<tr>
<th>converges</th>
<th>inx. ctrl</th>
<th>1e-12</th>
<th>1e-11</th>
<th>YES</th>
<th>YES</th>
<th>1e-10</th>
<th>NO</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMRES iter’s</td>
<td>2672</td>
<td>4020</td>
<td>3728</td>
<td>3404</td>
<td>&gt; 10000</td>
<td>&gt; 10000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CG iter’s</td>
<td>162</td>
<td>142</td>
<td>142</td>
<td>142</td>
<td>&gt; 500</td>
<td>&gt; 500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQP iter’s</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>&gt; 50</td>
<td>&gt; 50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As in Section 5.1, the inexactness control strategy outperforms the best fixed-tolerance guess, this time by about 20%, in terms of the number of GMRES iterations. (This is not unexpected, as the incomplete LU preconditioner is less effective when compared to Section 5.1, see Figure 5.12). Our algorithm requires one additional SQP iteration and 20 additional CG iterations, whose cost is offset by the low tolerance solves throughout the SQP run. Again, fast local convergence in the last few SQP iterations is evident, from Tables 5.5 and 5.6, although not yet supported by a theoretical convergence rate estimate.
Figure 5.12: Performance of preconditioned GMRES – Navier–Stokes.
<table>
<thead>
<tr>
<th>iter</th>
<th>$f(x_k)$</th>
<th>$|c(x_k)|$</th>
<th>$|\nabla_x L_k|$</th>
<th>$\Delta_k$</th>
<th>$n_k$</th>
<th>$t_k$</th>
<th>CG iter</th>
<th>accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000e+00</td>
<td>7.531966e-02</td>
<td>0.000000e+00</td>
<td>1.000000e+02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.12883e-01</td>
<td>2.390632e-02</td>
<td>7.154109e-02</td>
<td>1.000000e+02</td>
<td>3.266380e+00</td>
<td>2.561887e+00</td>
<td>17</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>3.49110e-01</td>
<td>7.949475e-03</td>
<td>1.105058e-01</td>
<td>1.000000e+02</td>
<td>1.468207e+00</td>
<td>1.752357e+00</td>
<td>19</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>3.49110e-01</td>
<td>7.949475e-03</td>
<td>1.105058e-01</td>
<td>1.538637e+00</td>
<td>6.962600e-01</td>
<td>3.077274e+00</td>
<td>19</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>3.09049e-01</td>
<td>4.470019e-03</td>
<td>1.820603e-01</td>
<td>1.538637e+00</td>
<td>6.962600e-01</td>
<td>1.538605e+00</td>
<td>6</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>1.82559e-01</td>
<td>7.790192e-04</td>
<td>1.389707e-02</td>
<td>5.495695e+00</td>
<td>5.904807e-01</td>
<td>5.172562e-01</td>
<td>17</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>1.80549e-01</td>
<td>2.076801e-05</td>
<td>7.733582e-04</td>
<td>5.495695e+00</td>
<td>1.266288e-01</td>
<td>7.545284e-02</td>
<td>19</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>1.80439e-01</td>
<td>1.489214e-08</td>
<td>4.471905e-06</td>
<td>5.495695e+00</td>
<td>1.871022e-03</td>
<td>5.266811e-03</td>
<td>19</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>1.80439e-01</td>
<td>4.740854e-13</td>
<td>1.714431e-06</td>
<td>5.495695e+00</td>
<td>3.620395e-06</td>
<td>6.180395e-03</td>
<td>20</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>1.80439e-01</td>
<td>1.417097e-13</td>
<td>1.383309e-08</td>
<td>5.495695e+00</td>
<td>3.646422e-11</td>
<td>1.858155e-06</td>
<td>15</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 5.5: SQP convergence history with inexactness control – Navier–Stokes.
<table>
<thead>
<tr>
<th>iter</th>
<th>(f(x_k))</th>
<th>(|c(x_k)|)</th>
<th>(|\nabla_x \mathcal{L}_k|)</th>
<th>(\Delta_k)</th>
<th>(|n_k|)</th>
<th>(|t_k|)</th>
<th>CG iter</th>
<th>accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000e+00</td>
<td>7.531966e-02</td>
<td>0.000000e+00</td>
<td>1.000000e+02</td>
<td></td>
<td></td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>1</td>
<td>1.12909e-01</td>
<td>2.389013e-02</td>
<td>7.184774e-02</td>
<td>1.000000e+02</td>
<td>3.266380e+00</td>
<td>2.572190e+00</td>
<td>17</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>3.46935e-01</td>
<td>7.848287e-03</td>
<td>1.090567e-01</td>
<td>1.000000e+02</td>
<td>1.467597e+00</td>
<td>1.726977e+00</td>
<td>19</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>3.46935e-01</td>
<td>7.848287e-03</td>
<td>1.090567e-01</td>
<td>1.528168e+00</td>
<td>6.871760e-01</td>
<td>3.056337e+00</td>
<td>19</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>3.08161e-01</td>
<td>4.413714e-03</td>
<td>1.762772e-01</td>
<td>1.528168e+00</td>
<td>6.871760e-01</td>
<td>1.528168e+00</td>
<td>6</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>1.82320e-01</td>
<td>7.825016e-04</td>
<td>1.404086e-02</td>
<td>5.94572e+00</td>
<td>5.838284e-01</td>
<td>5.458060e-01</td>
<td>17</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>1.80602e-01</td>
<td>2.018716e-05</td>
<td>4.951633e-04</td>
<td>5.94572e+00</td>
<td>1.322191e-01</td>
<td>6.554956e-02</td>
<td>18</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>1.80439e-01</td>
<td>2.627475e-08</td>
<td>3.693730e-06</td>
<td>5.94572e+00</td>
<td>2.200917e-03</td>
<td>6.485555e-03</td>
<td>19</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>1.80439e-01</td>
<td>2.868842e-10</td>
<td>5.904543e-12</td>
<td>5.94572e+00</td>
<td>6.585515e-06</td>
<td>6.203026e-03</td>
<td>20</td>
<td>Y</td>
</tr>
</tbody>
</table>

**Table 5.6:** SQP convergence history with exact solves – Navier–Stokes.
Chapter 6

Conclusion

Our work extends the design and the global convergence analysis of a class of trust-region SQP algorithms for smooth nonlinear optimization to allow for an efficient integration of inexact linear system solvers. Our approach does not rely on the estimation of Lipschitz constants and derivative norm bounds, and it does not require a basis–nonbasis decomposition of optimization variables. It extends the range of applicability of SQP algorithms with inexact linear system solves and allows for a rigorous integration of preconditioners for KKT systems in the SQP framework.

The main theoretical contribution is in the careful treatment of the tangential step computation, which often represents the most expensive algorithmic step in SQP methods based on the composite–step trust–region approach. For this purpose, we develop an inexact conjugate gradient algorithm that utilizes a series of approximate projections via Krylov subspace methods, whose accuracy is controlled by the SQP algorithm. The resulting SQP algorithm dynamically adjusts stopping tolerances for iterative linear system solves based on its current progress toward a KKT point. The stopping tolerances can be easily implemented and efficiently computed, and are sufficient to guarantee first–order global convergence of the algorithm.

The devised stopping tolerances for iterative linear system solvers are fully implemented in an SQP algorithm, whose performance is examined on optimal control problems governed by Burgers and Navier–Stokes equations. Global convergence is
observed. Additionally, our inexactness control strategy always outperforms the best possible fixed–tolerance guess, in terms of the total number of Krylov solver iterations, for the same quality of the solution. The local convergence behavior of our approach is similar to that of exact schemes, in which all linear systems are solved using a sparse direct solver.

It is rather interesting that our SQP algorithm exhibits excellent local convergence properties, even though the inexactness framework aims to only satisfy the global convergence requirements. A detailed theoretical investigation of the local behavior is therefore necessary. In addition, it would be interesting to test the performance of the algorithm on important large–scale problems in a parallel computing environment. Preliminary large–scale results obtained using a C++ implementation of a simplified version of the algorithm coupled with parallel Krylov solvers and overlapping KKT domain decomposition preconditioners are very encouraging.
Bibliography


the finite difference multigrid solution of an optimal control optimality system. 


for nonlinear optimization. In *Large-scale nonlinear optimization*, volume 83 of


